# Punktezählalgorithmen für den Hecke-Operator und Anwendungen auf Modulkurven von Geschlecht 4

Dem Fachbereich 6 (Mathematik und Informatik)
der Universität Duisburg-Essen
zur Erlangung des Doktorgrades
der Naturwissenschaften (Dr. rer. nat.)

im Juli 2007 vorgelegt von

Guido Blady

aus Essen

| | |
|---|---|
| Tag der Disputation: | 23.11.2007 |
| Prüfungsvorsitzender: | Prof. Dr. Rüdiger Schultz |
| 1. Gutachter: | Prof. Dr. Dr. hc. Gerhard Frey |
| 2. Gutachter: | Prof. Dr. Henning Stichtenoth (Istanbul) |

# Vorwort

Der Einsatz der Public-Key Kryptografie zur elektronischen Signatur und Verschlüsselung ist aus unserer heutigen, vernetzten Welt nicht mehr wegzudenken. Aufgrund der niedrigen Schlüssellängen erhalten Kryptosysteme, die auf Jacobischen von Kurven basieren, erhöhte Aufmerksamkeit und werden im Fall elliptischer Kurven bereits in Mobiltelefonen, Reisepässen und in vielen anderen Bereichen eingesetzt.

Will man ein Kryptosystem auf Jacobischen von Kurven aufbauen, ist es wichtig, dass die Gruppenordnung der $k$-rationalen Punkte der Jacobischen über dem endlichen Körper $k$ einen großen Primfaktor enthält. Das diskrete Logarithmusproblem kann sonst mit Hilfe des chinesischen Restsatzes zu schnell gelöst werden, was das Kryptosystem unbrauchbar machen würde. Wir benötigen also Methoden zur Bestimmung der $k$-rationalen Punkte der Jacobischen einer Kurve. Für Kurven von Geschlecht 1 bis 3 und hyperelliptische Kurven von Geschlecht 4 gibt es zahlreiche effiziente Methoden, deren Ansatz entscheidend von der Charakteristik $p$ des Grundkörpers $k$ beeinflußt wird.

Diese Methoden lassen sich leider nicht auf großes $p$ und nichthyperelliptische Kurven von Geschlecht 4 übertragen. Wir stellen in dieser Arbeit einen Algorithmus zur Bestimmung der $\mathbb{F}_p$-rationalen Punkte der Jacobischen von allgemeinen Modulkurven vor, der mit seiner linearen Komplexität in Zeit- und Speicheraufwand gerade für die zahlreichen nichthyperelliptische Modulkurven von Geschlecht 4 effizient ist. Wegen des Index-Calculus Angriffs von Diem (s. [Die05]) gelten Kurven von Geschlecht 4 erst ab 172-Bit großer Gruppenordnung als sicher. Wir zeigen in expliziten Beispielen, dass unser Algorithmus in dieser Grössenordnung praktikabel ist. Der Algorithmus ist voll parallelisierbar und in allen Eingabegrößen simultanisierbar.

In [PPW03] schlagen Pelzl, Wollinger und Paar hyperelliptische Kurven von Geschlecht 4 über Körpern mit 32-Bit für die Implementation von Kryptosystemen für ARM Prozessoren vor. Unser Algorithmus kann Beispiele dafür leicht berechnen und daher fügen wir dieser Arbeit im Anhang eine ganze Reihe davon hinzu.

**Danksagung**

Mein besonderer Dank gebührt meinem Doktorvater Herrn Prof. Dr. Dr. hc. Gerhard Frey für die Heranführung an dieses interessante Thema, für viele kreative Denkanstöße und für die unermüdliche Unterstützung bei der Fertigstellung dieser Arbeit.

Allen Doktoranden und Mitarbeitern des Instituts für Experimentelle Mathematik, insbesondere Holger Bleul, Björn Buth, Nihad Cosic, Dr. Thomas Dreibholz, Dr. Wolfgang Happle, Dr. Ingo Janiscak, Oscar Ledesma, Maxim Li, Marios Magioladitis, Dr. Roger Oyono, Xavier Taixés, Julia Thiemann und Marco Wolter möchte ich für das freundliche und hilfsbereite Umfeld danken, welches maßgeblich zu einem hervorragenden Arbeitsklima beigetragen hat.

Herrn Prof. Dr. Henning Stichtenoth möchte ich für seine Bereitschaft danken, diese Arbeit als Zweitgutachter anzunehmen.

Herrn Prof. Dr. Kumar Murty danke ich für die Gastfreundschaft während meines sechsmonatigen Forschungsaufenthalts an der Universtät Toronto von Juli bis Dezember 2003.

Ohne die finanzielle Unterstützung durch das Graduiertenkolleg der deutschen Forschungsgemeinschaft (DFG), „Mathematische und ingenieurwissenschaftliche Methoden für sichere Datenübertragung und Informationsvermittlung“ wäre diese Promotion nicht möglich gewesen.

Schliesslich gilt mein Dank meinen Eltern für ihren Glauben an mich und für ihre stete Unterstützung auf meinem Weg.

Essen, im Juli 2007 Guido Blady

# Inhaltsverzeichnis

# Kapitel 1

# Einleitung

Sei $C/k$ eine projektive irreduzible reguläre Kurve über dem Körper $k$ der Charakteristik $p \geq 0$. Für einen festen Punkt $P_0 \in C(\bar{k})$ wird durch die Abel-Jacobi Abbildung

$$\Theta_{P_0} : C(\bar{k}) \longrightarrow Pic^0(C)(\bar{k})$$

die Kurve $C$ in ihre Divisorklassengruppe $Pic^0(C)$ eingebettet. Hierbei wird jedem Punkt $P \in C(k)$ die Divisorklasse $[(P)-(P_0)]$ vom Grad 0 zugeordnet. Die Jacobische Varietät $J_C$ von $C$ ist eine abelsche Varietät mit $J_C \cong Pic^0(C)$. Die Dimension von $J_C$ stimmt mit dem Geschlecht der Kurve $C$ überein und falls der Basispunkt $P_0$ der Abel-Jacobi Abbildung ein $k$-rationaler Kurvenpunkt ist, so ist diese Abbildung ebenfalls über $k$ definiert.

Diese Relation zwischen $C$ und $J_C$ zeigt, dass sich Informationen über $C(k)$ aus $J_C(k)$ ableiten lassen und umgekehrt. Das Studium der jacobischen Varietät $J_C$ ist also ein wichtiges Werkzeug bei der Berechnung diophantischer Gleichungen, insbesondere der Beschreibung von $C(k)$.

Aus Sicht der Kryptografie sind wir vor allem an der Konstruktion von $J_C$ mit fast-primer Gruppenordnung $\#J_C(k)$ interessiert. Hier gibt es zwei gegensätzliche Ansätze: Man wählt $k$ fest und sucht eine geeignete Kurve $C|k$ oder man konstruiert eine „globale“ Kurve $C$ (z.B über $k = \mathbb{Q}$) und sucht geeignete Reduktionen modulo Primidealen $\mathfrak{P}$. Um die Bestimmung der Gruppenordnung der Jacobischen $J_C$ von $C$ modulo $\mathfrak{P}$ zu beschleunigen, benötigt man Informationen über den Endomorphismenring von $J_C$. Dies funktioniert bei Jacobischen mit komplexer Multiplikation und – wie wir sehen werden – mit reeller Multiplikation sehr gut.

Es entsteht also in beiden Fällen das Problem, die Ordnung $\mathcal{O}$ einer gegebenen Gruppe $J_C$ zu berechnen. Dazu gibt es vier verschiedene Strategien: Ein erster Ansatz zur Behandlung dieses Problems ist die Anwendung von *generischen Algorithmen*, mit denen alle Gruppen behandelbar sind. Sie berechnen gleichzeitig das diskrete Logarithmusproblem und die Gruppenordnung und haben exponentielle Komplexität. Trotzdem sind sie nützlich, wenn man bereits einige Information über die Gruppenordnung besitzt, siehe Beispiel 2.5.4.

Die anderen drei Strategien basieren auf der Berechnung des charakteristischen Polynoms $\chi_\phi(T)$ des Frobenius-Endomorphismus $\phi$. Es gilt

$$\#J_C(k) = \chi_\phi(1).$$

Die *l-adischen Methoden* approximieren $\chi_\phi(T)$ auf den Tate-Modul zu Primzahlen $l \neq \text{char } k > 0$ die relativ klein sind und verwenden dann die Sätze von Weil zur Bestimmung des charakteristischen Polynoms mit Hilfe des chinesischen Restsatzes. Die *p-adischen Methoden* approximieren das charakteristische Polynom auf der $p$-adischen Kohomologie. Beide Methoden erzielen praktikable Algorithmen für Kurven bis Geschlecht 3, falls für die Charakteristik $p \leq 5$ gilt. Für hyperelliptische Kurven gibt es eine $p$-adische Methode von HARVEY, die es ermöglicht, $\chi_\phi(T)$ in $O(\sqrt{p})$ Schritten zu berechnen, s. [Hav06]. In Kapitel 2 geben wir einen Überblick über die etablierten generischen, $l$-adischen und $p$-adischen Methoden.

Ein für die Kryptografie besonders interessanter Fall ist es, wenn $k$ ein Primkörper und die Charakteristik $p$ daher sehr groß ist. Für großes $p$ können wir außer der $p$-adischen Methode von HARVEY die *Endomorphismen-Methoden* verwenden: Die Gruppenordnung $\#J_C(k)$ wird dabei aus der Darstellungsmatrix eines geeigneten Endomorphismus errechnet. Hier steht uns der Cartier-Operator als Dual des Frobenius zur Verfügung, um Informationen modulo $p$ über die Gruppenordnung zu erhalten. Daher ist es interessant, schnelle Berechnungsmethoden für den Cartier-Operator zu untersuchen. Der Cartier-Operator wird durch die Hasse-Witt Matrix dargestellt, für welche wir eine Beschreibung in einer generischen Basis liefern, die BOUW in [Bou98] für Quartiken verwendet hat und die wir in dieser Arbeit auf projektive Kurven verallgemeinern. Dies wird im Abschnitt 2.5 behandelt.

Da der Cartier-Operator auf hyperelliptischen Kurven nur auf univariaten Polynomen agiert, kann man daraus ein effizientes Verfahren zur Berechnung der Hasse-Witt Matrix konstruieren (s. [BGS04]) und damit das charakteristische Polynom $\chi_\phi(T)$ mod $p$ berechnen. Dieses Verfahren verallgemeinern wir in dieser Arbeit durch Verwendung des Lagrange-Interpolationspolynoms auf bivariate Polynome und damit auf nicht-hyperelliptische Kurven. Für Kurven von Geschlecht 1, 2 und 3 resultiert diese Methode in effizienten Algorithmen. Für höheres Geschlecht zeigt sich allerdings, dass der nachfolgende Schritt, mit Hilfe eines Baby-Step/Giant Step Algorithmus das charakteristische Polynom genau zu bestimmen, eine höhere Komplexität als lineare Algorithmen hat, s. Bemerkung 2.5.5. Insbesondere hat dieser Schritt für Kurven von Geschlecht 4 eine Komplexität von $O(p^{\frac{5}{4}})$.

Um die wichtigen Fälle von Kurven von Geschlecht $2 \leq g \leq 4$ behandeln zu können, benötigt man speziellere Endomorphismenringe. Der Hauptteil der Arbeit beschäftigt sich in den übrigen Kapiteln mit dem Fall, dass $J_C$ ein Faktor der Jacobischen einer Modulkurve $X_0(N)$ ist. In diesem Fall gilt

$$\#J_C(\mathbb{F}_p) = \chi_{T_p}(p+1),$$

wobei $\chi_{T_p}$ das charakteristische Polynom des Hecke-Operators $T_p$ ist. Wir stellen einen Algorithmus zur Berechnung der Darstellungsmatrix des Hecke-Operators und damit Errechnung des charakteristischen Polynoms $\chi_{T_p}$ von $J_C$ vor. Dieser Algorithmus hat eine Komplexität von $O(p)$ in Zeit- und Speicherbedarf, und hat daher praktische Relevanz für nichthyperelliptische Modulkurven von Geschlecht 4 und große Charakteristik $p$.

In Kapitel 3 führen wir die begrifflichen und strukturellen Grundlagen für die Definition des Hecke-Operators $T_p$ ein. Wir definieren die Modulkurve $X_0(N)$, die durch die Bahnen der Operation von der Kongruenzuntergruppe $\Gamma_0(N)$ auf der erweiterten oberen Halbebene $\mathbb{H}^*$ induziert wird. Dann definieren wir die Modulformen $M_k(N)$, zeigen dass diese eine Darstellung als Fourierreihe $f(x) = \sum_i a_i q^i$ besitzen und beschreiben die Isomorphie zwischen den Spitzenformen $S_2(N)$ und den holomorphen Differentialen $\Omega^1(X_0(N))$. Danach führen wir die Hecke-Algebra $\mathbb{T}_N$ ein und beschreiben die Operation von $T_n$ auf den Modulformen, die letztlich eine Basisberechnung der Spitzenformen ermöglicht. Der Rest des Kapitels untersucht die Arithmetik der Modulsymbole $\mathcal{M}_2(N)$ und gibt eine konkrete Beschreibung des Hecke-Operators $T_n$ auf $\mathcal{M}_2(N)$. Diese Operation ist gegeben durch die Aktion einer Teilmenge $U$ von $\mathrm{Mat}^{2\times 2}(\mathbb{Z})$ die der Bedingung $C_n$ genügt:

$$\sum_{M\in U} u_M((\infty)M - (0)M) = (\infty) - (0)$$

und $\det M = n$ für alle $M \in U$. Diese Matrizenmengen bilden die Grundlage für konkrete Rechnungen in Kapitel 4.

In Kapitel 4 stellen wir eine effiziente Implementierung des Hecke-Operators $T_p$ für eine Primzahl $p$ vor. Basmaji hat in seiner Dissertation einen Algorithmus vorgestellt, der auf einer Idee von Merel basiert. Dieser Algorithmus verwendet eine Menge $\overline{S}_p$ der Mächtigkeit $\tilde{O}(p)$, die der Bedingung $C_p$ genügt. Wir geben eine bijektive Darstellung dieser Menge, um fehlerhafte Verdopplungen bei der Anwendung der Matrizen zu vermeiden. Der Vorzug von Basmajis Algorithmus ist die Möglichkeit einer simultanen Berechnung mehrer Hecke-Operatoren $T_{p_1}, \ldots, T_{p_l}$ mit $p_1 \equiv p_2 \equiv \ldots \equiv p_l$ modulo $N$. Wir zeigen, wie man die Kongruenzbedingung sogar fallen lassen kann und beschreiben ein Verfahren, mit dem man die Operatorensequenz zusätzlich simultan für mehrere Stufen berechnen kann, was z.B. bei der Suche nach Kongruenzen zwischen Modulformen nützlich ist, wie es in der Arbeit von Teixés benötigt wird, s. [Tei07]. Wie der Algorithmus von Basmaji ist auch unserer Algorithmus voll parallelisierbar, was wir an einigen Beispielen mit Kurven von Geschlecht $g = 4$ für Primzahlen mit 43 Bit zeigen. Desweiteren werden einige kleinere Korrekturen gegenüber Basmajis Algorithmus und hilfreiche Bemerkungen für die Implementation aufgeführt und die Leistung des Algorithmus in der Theorie und Praxis untersucht.

Die Aufzählung von $\overline{S}_p$ geschieht hauptsächlich durch die Generierung teilerfremder Tupel, was wir durch die Einführung eines Siebverfahrens in Kapitel 5 optimieren. Wir stellen den konventionellen Methoden der euklidischen Algorithmenvariationen ein effizientes Siebverfahren gegenüber, welches aus einer Liste von Primteilern zu gegebener Zahl $x$ alle teilerfremden Zahlen $y$ aussiebt, ähnlich wie beim Sieb des Erathostenes. Dazu beschreiben wir eine speichereffiziente, schnell zugängliche Organisation der Tabelle der Teiler und zeigen experimentell, dass dieses Verfahren durchschnittlich $2,5$ mal schneller als die euklidischen Algorithmusverfahren ist.

In Anhang A finden sich jeweils 100 Beispiele zu charakteristischen Polynomen $\chi_{T_p}$ von Hecke-Operatoren $T_p$ mit $p \sim 2^{32}$ prim zu den Modulkurven

der Stufen $N = 23, 47$ und 53, um die statistische Verteilung großer Primteiler von $\#J_0(N)(\mathbb{F}_p)$ zu veranschaulichen. In Anhang B werden einige Beispiele zu Matrizen von Hecke-Operatoren $T_p$ mit $p \sim 2^{43}$ prim zu den Modulkurven der Stufen $N = 47$ und 53 aufgeführt. Die zugrundeliegenden Algorithmen wurden in `C++` programmiert, wobei die Modulsymbolreduktion mit `Magma` erfolgte.

# Kapitel 2

# Punktezählalgorithmen

Will man jacobische Varietäten über einem endlichen Körper in der Kryptografie nutzen, so ist ein wichtiges Kriterium für die Sicherheit, dass die Gruppenordnung der Varietät über dem endlichen Körper einen großen Primzahlteiler hat, damit das *diskrete Logarithmusproblem* schwer zu lösen ist.

Die Bestimmung dieser Gruppenordnung ist also ein wesentlicher Bestandteil bei der Erstellung von Kryptosystemen, die auf jacobischen Varietäten basieren. Will man Jacobische mit komplexer Multiplikation verwenden, so kann man zu einer vorgegebenen geeigneter Ordnung $\mathcal{O}$ eines CM-Körpers eine Jacobische über $\mathbb{Q}$ mit einem Endomorphismenring konstruieren, der isomorph zu dieser Ordnung ist. Für jede Primzahl $p \in \mathbb{Z}$, die in $\mathcal{O}$ zerlegt ist, erhält man dann die Gruppenordnung der Jacobischen über $\mathbb{F}_p$. Hier gibt es effiziente Algorithmen für elliptische Kurven und hyperelliptische Kurven von Geschlecht 2 und 3, siehe das umfangreiche Buch [CoF$^+$06] von Cohen, Frey und weiteren Autoren.

Im Allgemeinen steht man vor der Situation, dass eine zufällige Kurve über einem endlichen Körper $k$ gegeben ist. Wir unterscheiden dann vier verschiedene Strategien: *Generische Methoden* für das diskrete Logarithmusproblem in allgemeinen Gruppen werden im ersten Abschnitt besprochen. Die anderen drei Methoden basieren auf der Berechnung des charakteristischen Polynoms des Frobenius-Endomorphismus auf geeigneten Darstellungsräumen. Dies erfolgt mit den sogenannten *$l$-adischen Methoden* auf dem Tate-Modul, mit den *$p$-adischen Methoden* auf der $p$-adischen Kohomologie, oder mit *Endomorphismen-Methoden*, bei denen man im allgemeinen Fall den Cartier-Operator ausnutzen wird.

## 2.1 Generische Methoden

Im folgenden werden nun drei generische Algorithmen und deren Aufwand betrachtet. Fortlaufend durch diesen Abschnitt sei $(G, 0, \oplus)$ eine endliche, zyklische Gruppe mit Ordnung $n$ die von einem Element $g$ erzeugt wird. Für $m \in \mathbb{Z}$ und $a \in G$ bezeichne $[m]a$ die $m$-fache Addition eines Elementes $a$ zu sich selbst. Der diskrete Logarithmus $\log_g a$ bedeutet dann die Angabe einer Zahl $l \in \mathbb{Z}$, so dass gilt: $a = [l]g$. Dabei ist $l$ eindeutig modulo $n$ bestimmt. Für

die Anwendung der Algorithmen ist es nur erforderlich, dass $G$ eine *Black Box Gruppe* ist. Das heißt, dass man in $G$ addieren und invertieren kann, und dass zwei Elemente aus $G$ auf Gleichheit getestet werden können. Ausgehend von $a$ und $g$ wird dann der diskrete Logarithmus mit diesen elementaren Operationen gelöst.

Das Berechnen der Gruppenordnung ist der Spezialfall $a = 0 = [l]g$. In den nachfolgenden Algorithmen gehört die Gruppenordnung $n$ zwar zum Eingabeparameter, man arbeitet dann aber einfach mit einer Obergrenze für $n$. Wenn selbst so etwas nicht verfügbar sein sollte, nimmt man irgendeinen Wert für $n$ an – wird dadurch der diskrete Logarithmus nicht gefunden, erhöht man $n$ so lange, bis der Algorithmus terminiert.

### 2.1.1 Der Baby-Step/Giant-Step Algorithmus

Der folgende Algorithmus wurde 1971 von SHANKS publiziert und wurde als *Baby-Step/Giant-Step Algorithmus* bekannt. War er ursprünglich dazu gedacht, Klassenzahlen von imaginärquadratischen Zahlkörpern zu berechnen, gibt es viele Varianten, wie z.B. die der Bestimmung der Punktezahl einer elliptischen Kurve oder wie in unserem Fall, zur Berechnung des diskreten Logarithmus. Fast wie bei dem naiven Ansatz werden hier alle Vielfachen von $g$ auf Gleichheit mit $a$ getestet, nur dass die Effizienz durch Zwischenspeichern auf einer passenden Datenstruktur erhöht wird.

Anstatt $g, [2]g, ...$ usw. zu berechnen, bis $a$ gefunden ist, wird eine kleinere „Baby-Step"-Liste erstellt. Dann berechnet man für ein passendes $m \in \mathbb{N}$ solange die „Giant-Steps" $[m]g, [2m]g, ...$ bis man ein $[mi]g$ gefunden hat, welches in der Baby-Stepliste vorkommt. Würde man $[mi]g$ sukzessive in der Baby-Step Liste suchen, böte dieser Algorithmus keinen Vorteil gegenüber dem naiven Ansatz. Hat jedoch jedes Element aus $G$ einen eindeutigen Repräsentanten, und lassen sich diese anordnen, dann kann man die Baby-Step Liste entsprechend erstellen und mit einer binären Suche bearbeiten.

Damit ist eine weitere Anforderung an $G$ gestellt, die jedoch in der Praxis von den meisten Gruppen erfüllt wird: In $\mathbb{F}_p^\times$ hat jedes Element einen eindeutigen Repräsentanten in $\{1, ..., p-1\}$ und man kann dessen natürliche Ordnung verwenden. In $\mathbb{F}_{p^s}^\times$ kann man jedes Element eindeutig durch ein Polynom aus $\mathbb{F}_p[X]$ darstellen, dessen Grad kleiner als $s$ ist. Damit kann man die Elemente erst nach Grad des Polynoms, dann nach den Koeffiezienten ordnen. Im Fall einer elliptischen Kurve über einem endlichen Körper sind die Punkte Elemente aus $\mathbb{F}_q \times \mathbb{F}_q$ und können lexikografisch nach den zwei Komponenten geordnet werden.

---

**Algorithmus 2.1.1** Baby-Step/Giant-Step Algorithmus

**Eingabe:** Eine endliche zyklische Gruppe $G = \langle g \rangle$ mit $\#G = n$ und $a \in G$.
**Ausgabe:** Eine ganze Zahl $l$ mit $a = [l]g$.

1. Wähle eine Anzahl Baby-Steps $m$. Berechne die Paare $([i]g, i)$ für $0 \leq i < m$ und speichere sie in einer nach der ersten Komponente geordneten Liste.

2. Die Größe der Giant-Steps ist $k = \lceil \frac{n}{m} \rceil$. Berechne $a \oplus [-jm]g$ für $0 \leq j < k$ und versuche $a \oplus [-jm]g$ mit einer binären Suche in der Liste der vorberechneten $[i]g$ zu finden.

3. Ist das Element in der Liste vorhanden, berechne $\log_g a = jm + i$.

---

Es ist gesichert, dass der Algorithmus terminiert, da $\log_g a$ ein Element aus $\{0, ..., n-1\}$ ist und damit eine eindeutige Darstellung $\log_g a = jm + i$ mit $0 \leq i < m$ und $0 \leq j < \lceil \frac{n}{m} \rceil$ hat. Die Berechnung der Baby-Steps und das Sortieren der Liste benötigt $O(m \log m)$ Operationen, wobei eine Operation einer Addition und dem Vergleich zweier Gruppenelemente entspricht. Die Giant-Steps benötigen $O(k \log m)$ Operationen und liegen damit von den Koeffizienten her etwa in der selben Größenordnung. Um die Anzahl der Baby-Steps zu optimieren, vernachlässigen wir also den Term $\log m$ und minimieren $m + k$ unter der Bedingung $mk = n$. Also gilt $m = k = \lceil \sqrt{n} \rceil$ und die gesamte Zeitkomplexität beläuft sich auf $O(\sqrt{n} \log n)$. Dabei wird Speicherplatz in der Komplexität von $O(\sqrt{n} \log n)$ benötigt, um $\lceil \sqrt{n} \rceil$ Einträge der Länge $O(\log n)$ zu speichern.

### 2.1.2 Pollards $\rho$-Methode

Ein Nachteil von Algorithmus 2.1.1 ist der Bedarf an Speicherplatz, wenn man mit großen Gruppen arbeitet. Pollard veröffentlichte daher 1978 einen probabilistischen Algorithmus mit etwa der selben Laufzeit, der jedoch praktisch ohne Speicherplatzbedarf auskommt.

---

**Algorithmus 2.1.2** Pollards $\rho$-Methode

**Eingabe:** Eine endliche zyklische Gruppe $G = \langle g \rangle$ mit $\#G = n$ und $a \in G$.
**Ausgabe:** Eine ganze Zahl $l$ mit $a = [l]g$.

1. Teile $G = T_1 \dot\cup T_2 \dot\cup T_3$ in eine zufällige Zerlegung dreier Partitionen auf und bilde $x_0 = [k_0]g \oplus [m_0]a$ für zufällig gewählte Zahlen $k_0$ und $m_0$.

2. Bilde rekursiv die Folgen $(x_i)_{i \geq 0}$, $(k_i)_{i \geq 0}$ und $(m_i)_{i \geq 0}$ über

$$x_{i+1} = \begin{cases} a \oplus x_i & : \quad x_i \in T_1 \\ [2]\, x_i & : \quad x_i \in T_2 \\ g \oplus x_i & : \quad x_i \in T_3 \end{cases} \qquad k_{i+1} = \begin{cases} k_i & : \quad x_i \in T_1 \\ 2k_i & : \quad x_i \in T_2 \\ k_i + 1 & : \quad x_i \in T_3 \end{cases}$$

$$m_{i+1} = \begin{cases} m_i + 1 & : \quad x_i \in T_1 \\ 2m_i & : \quad x_i \in T_2 \\ m_i & : \quad x_i \in T_3 \end{cases}$$

Dann gilt $x_i = [k_i]g \oplus [m_i]a$ für alle $i \geq 0$.

3. Da G endlich ist, wird die Folge $(x_i)$ irgendwann periodisch und es existieren ganze Zahlen $\mu \geq 0$ und $\lambda \geq 1$, so dass $x_1, ..., x_{\mu+\lambda-1}$ paarweise verschieden sind und $x_{i+\lambda} = x_i$ für $i \geq \mu$.

4. Hat man also ein $x_i = x_j$ für $i \neq j$ gefunden, dann gilt $[k_i + lm_i]g = [k_j + lm_j]g$ und damit

$$l(m_j - m_i) \equiv k_i - k_j \pmod{n} \tag{2.1}$$

Diese Gleichung kann man lösen, falls $d = \text{ggT}(n, m_j - m_i) = 1$.

5. Anderenfalls hat man $d$ mögliche Werte für $l$, die man für $[l]g = a$ bei kleinem $d$ schnell testen kann: Man berechnet $d = un + v(m_j - m_i)$ mit Hilfe des erweiterten euklidischen Algorithmus für ganze Zahlen $u$ und $v$. Multipliziert man (1.1) mit $v$ erhält man daher

$$ld \equiv v(k_i - k_j) \pmod{n}$$

Damit gilt also $d|(k_i - k_j)$ und $l$ ist gleich einem der $d$ Werte

$$\frac{v(k_i - k_j)}{d} + s\frac{n}{d} \pmod{n} \qquad \text{mit } 0 \leq s < d$$

---

Pollards $\rho$-Methode erhielt ihren Namen dadurch, dass die graphische Darstellung von $x_1, ..., x_\mu$ als Linie, verbunden mit $x_\mu, ..., x_{\mu+\lambda-1}$ als Kreis, an den griechischen Buchstaben $\rho$ erinnert.

Der erstmögliche Treffer ist $x_\mu = x_{\mu+\lambda}$. Das heißt es müssen mindestens $\mu + \lambda$ Folgeglieder berechnet werden. Die Abbildung

$$F : G \longrightarrow G, \quad x \mapsto \begin{cases} ax_i & : \quad x_i \in T_1 \\ x_i^2 & : \quad x_i \in T_2 \\ gx_i & : \quad x_i \in T_3 \end{cases}$$

verhält sich als Zufallsfunktion etwa gleichverteilt und daher ist der Erwartungswert von $\mu + \lambda$ etwa $1,25\sqrt{n} \in O(\sqrt{n})$ Gruppenoperationen (siehe [CoF$^+$06]).

### 2.1.3 Index Calculus

Wir wollen zuerst den Begriff der subexponentiellen Komplexität erläutern: Betrachte die *Komplexitätsfunktion*

$$L_x(u, v) = \exp((v + o(1)) \log(x)^u \log(\log(x))^{1-u}),$$

mit $0 \leq u \leq 1$ und $v > 0$. $o(1)$ bezieht sich auf das asymptotische Verhalten von $x$. Hier ist $u$ der zentrale Parameter, da $L_x(u, v)$ zwischen polynomieller ($u = 0$) und exponentieller ($u = 1$) Komplexität interpoliert. Für $u < 1$ nennt man die Komplexität *subexponentiell*.

Sei $p$ der grösste Primteiler der Gruppenordnung $n$ von $G$. Nach einem Satz von SHOUP können Algorithmen mit einer Laufzeit besser als $O(\sqrt{p})$ nicht generisch sein (s. [CoF$^+$06], Kap. 19). Dies gilt nicht nur für Methoden von subexponentieller Laufzeit, sondern auch für jene Methoden, deren Komplexität exponentiell und von der Form $O((\#G)^\alpha)$ mit $\alpha < 1/2$ sind.

In der Anwendung arbeitet man immer mit einer konkreten Darstellung einer gegebenen Gruppe $G$ und viele Arten von Gruppen haben gemeinsame Eigenschaften. Daher gibt es Methoden, die bis auf bestimmte Eigenschaften allgemein beschrieben werden können und erheblich bessere Komplexitäten aufweisen, als die besten generischen Algorithmen. Für manche dieser Methoden erhält man subexponentiellen Zeitaufwand und für manche Methoden erhalten wir sogar exponentielle Laufzeiten mit $\alpha < 1/2$, wie z.B. jene, die auf den Jacobischen von hyperelliptischen Kurven mittleren Geschlechts basieren. Diese Algorithmen gehören zu der Familie der *Index Calculus Algorithmen*.

Die Index Calculus Algorithmen basieren auf der folgenden Idee: Sei wieder $G$ eine zyklische Gruppe der Ordnung $n$ und $g \in G$ eine erzeugendes Element von $G$. Gilt

$$\bigoplus_{i=1}^{r} [e_i] g_i = 0 \tag{2.2}$$

für Elemente $g_i \in G$ und passende Vielfache $e_i$ dann folgt

$$\sum_{i=1}^{r} e_i \log_g(g_i) \equiv 0 \bmod n. \tag{2.3}$$

Falls wir viele Gleichungen der Form (2.2) erstellen können und mindestens eine davon ein Element enthält, für das der diskrete Logarithmus bekannt ist, dann können wir das System (2.3) für die Einträge $\log_g(g_i)$ lösen, falls das System nicht zu groß ist. Die Menge $\{g_1, \ldots, g_r\}$ nennt man eine *Faktorbasis*. Fügen wir der Faktorbasis ein Element $a$ mit $a = [l]g$ für unbekanntes $l$, so können wir hoffen, auf diese Weise $l$ zu ermitteln.

Eine ausreichende Menge an Gleichungen der Form (2.2) zu erstellen ist äquivalent dazu, die Struktur von $G$ als $\mathbb{Z}$-Modul zu berechnen. Sei $\mathbb{Z}^r$ die freie abelsche Gruppe, die erzeugt wird durch die Elemente $\{A_i, \ldots, A_r\}$ und $L$ das Gitter in $\mathbb{Z}^r$, dass erzeugt wird durch die Relationen $\sum_{i=1}^{r} e_i \cdot A_i = 0$, mit $e_i$ aus den Gleichungen (2.2). Wir erhalten dann einen Homomorphismus

$$\begin{array}{rcl} \Phi : \mathbb{Z}^r & \longrightarrow & G \\ (e_1, \ldots, e_r) & \mapsto & \displaystyle\bigoplus_{i=1}^{r} [e_i] g_i \end{array}$$

mit Kern $L$, so dass $\mathbb{Z}^k / L \simeq G$.

Für eine passende Wahl der Faktorbasis muß man idealerweise zeigen, dass ein zufällig gewähltes Element aus $G$ mit hoher Wahrscheinlichkeit als Linearkombination einer kleinen Anzahl an Elementen aus der Faktorbasis und mit

kleinen Potenzen dargestellt werden kann. Die subexponentielle Komplexität der Index Calculus Methoden hängt von der Effizienz ab, mit der diese Relationen generiert werden können, da dies meist der aufwendigste Teil des Algorithmus ist.

### Arithmetische Formationen

**Definition 2.1.3** *Sei $\mathcal{P}$ eine abzählbare Menge, deren Elemente im folgenden **Primelemente** genannt werden. Eine **additive arithmetische Halbgruppe** ist ein freier abelscher Monoid $\mathcal{M}$ über $\mathcal{P}$ zusammen mit einer Äquivalenzrelation $\sim$, die kompatibel mit dem Kompositionsgesetz ist und mit der gilt $G \simeq \mathcal{M}/\sim$.*

*Jedes Element $g \in G$ wird eindeutig durch ein Element $\imath(g) \in \mathcal{M}$ dargestellt derart, dass der Isomorphismus $G \to \mathcal{M}/\sim$ durch $g \mapsto \imath(g)/\sim$ gegeben ist.*

*Eine **Größenabbildung** ist eine Norm, die als Monoid-Homomorphismus $(\mathcal{M}, \oplus) \to (\mathbb{R}, +)$ gegeben ist und die vollständig durch die Werte an den Primelementen von $\mathcal{M}$ bestimmt ist. Wir setzen voraus, dass alle Primelemente $p \in \mathcal{P}$ eine positive Größe haben. Diese Abbildung kann durch Verwendung von $\imath$ auch auf $G$ angewendet werden. Die Größe von $g \in G$ (bwz. $m \in \mathcal{M}$) wird mit $|g|$ (bzw. $|m|$) bezeichnet.*

*Eine solche Gruppe $G$, zusammen mit dem Monoid $\mathcal{M}$, der Äquivalenzrelation $\sim$, der Darstellungsabbildung $\imath$ und der Größenabbildung, also das Quintupel $(G, (\mathcal{M}, \oplus), \sim, \imath, |\cdot|)$ wird dann eine **arithmetische Formation** genannt.*

Wir nehmen an, dass die Elemente von $G$ durch Bitstrings der korrespondierenden Elemente von $\mathcal{M}$ dargestellt sind und die Länge dieser Bitstrings durch ein $r$ nach oben abgeschätzt ist. Desweiteren gehen wir davon aus, dass alle generischen Operationen in $G$ in polynomialer Zeit in $r$ durchgeführt werden.

**Definition 2.1.4** *Eine **Glattheitsgrenze** $B$ ist eine natürliche Zahl und wir bezeichnen mit $\mathcal{M}_B$ (bzw. $\mathcal{P}_B$) die Menge der Elemente von $\mathcal{M}$ (bzw. $\mathcal{P}$) deren Größe kleiner oder gleich $B$ ist. $n_B$ sei die Mächtigkeit von $\mathcal{P}_B$ und $n'_B$ sei die Mächtigkeit von $\mathcal{M}_B$. Ein Element $g$ von $G$ wird $B$**-glatt** genannt, falls die Zerlegung von $g$ in $\mathcal{M}$ nur aus Primelementen in $\mathcal{P}_B$ besteht.*

Nach Möglichkeit wird die Faktorbasis durch eine Glattheitsgrenze definiert und als $\mathcal{P}_B$ dargestellt. In unserem Kontext benötigen wir, dass $n'_B$ endlich und eine polynomielle Darstellung in $B$ hat. Die Elemente von $\mathcal{M}_B$ sollen mit einem Zeitaufwand aufgezählt werden können, der polynomiell in $B$ und linear in $n'_B$ ist. Wir benötigen weiterhin, dass ein Element $m \in \mathcal{M}$ auf Enthaltensein in $\mathcal{P}$ in polynomieller Zeit in $|m|$ und linearer Zeit in $n'_{|m|}$ getestet werden kann. Dies kann zum Beispiel durch Probedivision aller Elemente mit Norm kleiner als $|m|$ durchgeführt werden. $\mathcal{P}_B$ kann also in polynomieller Zeit in $B$ und quadratischer Zeit in $n'_b$ konstruiert werden. Es sollte in $O(n'_B)$ Operationen möglich sein, Elemente von $G$ auf $B$-Glattheit zu testen und diese in Primelemente zu zerlegen. Dies kann in der Praxis sogar noch schneller durchgeführt werden. Wir geben nun ein Beispiel für eine arithmetische Formation. Für weitere Beispiele, siehe [CoF$^+$06], Kapitel 20.

**Beispiel 2.1.5** Sei $C$ eine hyperelliptische Kurve von Geschlecht $g$ über einem endlichen Körper $K$ der Charakteristik $p$ und $G$ die Divisorklassengruppe $G = Pic_C^0(K)$. Hier ist $\mathcal{P}$ die Menge der irreduziblen Divisoren. Wir wissen, dass sich jedes Element von $G$ durch einen $K$-rationalen Divisor mit einem Grad von höchstens $g$ darstellen lassen kann.

Die Menge der Primelemente ist definiert durch die Menge der Primdivisoren, deren effektive Divisoren irreduzibel über $K$ sind. Letztere sind entweder einzelne $K$-rationale Punkte, oder Summen von Galois-Konjugierten über $K$ eines nicht-$K$-rationalen Punktes. Mit anderen Worten: falls der Divisor $D$ eine Mumford-Darstellung $[u_D, v_D]$ besitzt, ist $D$ ein Primelement genau dann, wenn das Polynom $u_D$ irreduzibel über $K$ ist. Die Grösse des Divisors ist der Grad von $u_D$.

Eine mögliche Beschreibung des Index Calculus Algorithmus in seiner grundlegenden Form ist die folgende. Tatsächlich gibt es einige Varianten, aber diese weichen nur in kleineren Details voneinander ab.

---

**Algorithmus 2.1.6** Index Calculus

**Eingabe:** Eine Gruppe $G$ der Ordnung $n$ und Elemente $g, a \in G$ mit $a \in \langle g \rangle$.
**Ausgabe:** Eine ganze Zahl $l$ mit $a = [l]g$.

1. **Konstruktion der Faktorbasis**
   Wähle einen Glattheitsgrenze $B$ und setze dazu als Faktorbasis die Menge $\mathcal{P}_B = \{\pi_1, \ldots, \pi_{n_B}\}$ der $B$-glatten Primelemente von $G$.

2. **Erstellen der Relationen**
   Dies sind die Gleichungen der Form

   $$[x_i]g \oplus [y_i]a = \bigoplus_{j=1}^{n_B} [e_{i,j}]\pi_j \quad \text{für } i = 1, 2, \ldots$$

   Sei $c = n_B$. Konstruiere eine Matrix $A$ mit $c$ Zeilen, die durch die Zeilenvektoren

   $$(e_{i,1}, e_{i,2}, \ldots, e_{i,c}) \quad \text{für } i = 1, 2, \ldots, c$$

   definiert ist. Speichere die Vektoren $\mathbf{x} = (x_1, x_2, \ldots, x_c)$ und $\mathbf{y} = (y_1, y_2, \ldots, y_c)$ und setze

   $$\mathbf{z} = (e_{c+1,1}, e_{c+1,2}, \ldots, e_{c+1,c}).$$

   Falls möglich, bearbeite die Matrix $A$ und die Vektoren $\mathbf{x}$, $\mathbf{y}$ und $\mathbf{z}$, um $c$ zu verringern.

3. **Lineare Algebra**
   Berechne eine Lösung $\mathbf{s}A \equiv \mathbf{z} \bmod n$, oder finde ein Element $\mathbf{s}$ aus dem Kern von $A$, also $\mathbf{s}A = 0 \bmod n$.

4. **Ermitteln der Lösung**

   Nach Konstruktion von $A$ und den Vektoren $\mathbf{x}$ und $\mathbf{y}$ gilt die folgende formale Relation in $G$. (Dabei bedeuted $(\cdot)^t$ Transposition):

$$(\mathbf{x}^t\mathbf{y}^t) \times \begin{bmatrix} g \\ a \end{bmatrix} = A \times \Pi \quad \text{mit} \quad (\mathbf{x}^t\mathbf{y}^t) = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_c & y_c \end{bmatrix} \text{ und } \Pi = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_c \end{bmatrix}.$$

   Sei $\mathbf{s} = (s_1, \ldots, s_c)$ der Vektor aus Schritt 3 und sei $\alpha$ bzw. $\beta$ gleich dem Skalarprodukt $\alpha = \mathbf{s}\mathbf{x}^t$, bzw. $\beta = \mathbf{s}\mathbf{y}^t$.

   (a) Falls $\mathbf{s}A = 0$, dann gilt $\mathbf{s}A\Pi = 0$ und wir erhalten $(\mathbf{x}^t\mathbf{y}^t) \times \begin{bmatrix} g \\ a \end{bmatrix} = 0$, das heißt $[\alpha]g \oplus [\beta]a = 0$. Also ist $\log_g(a) = -\frac{\alpha}{\beta} \bmod n$, vorausgesetzt es gilt $\mathrm{ggT}(\beta, N) = 0$.

   (b) Falls $\mathbf{s}A = \mathbf{z}$, dann gilt

$$[\alpha]g \oplus [\beta]a = \mathbf{s}A\Pi = \mathbf{z}\Pi = [x_{c+1}]g \oplus [y_{c+1}]a$$

   und daraus folgt $\log_g(a) = -\frac{\alpha - x_{c+1}}{\beta - y_{c+1}} \bmod n$.

---

Es gibt viele Möglichkeiten, die einzelnen Schritte von Algorithmus 2.1.6 zu verbessern, siehe [CoF$^+$06], Kapitel 20. Für die Analyse einer Index Calculus Variante müssen einige Probleme in Betracht gezogen werden. Falls $B$ zu klein ist, wird es wahrscheinlich zu lange dauern, geeignete Relationen zu finden. Ist $B$ hingegen zu groß, wird der lineare Algebra-Schritt zu teuer. Die Konstruktion der Faktorbasis hat eine Laufzeit von höchstens $\tilde{O}((n'_B)^2) \subset \tilde{O}((n_B)^2)$, wenn man die Elemente mit Norm $B$ aufzählt und testet, ob diese Primelemente sind. Für das Erstellen der Relationen muss man $\tilde{O}(n'_B \cdot \#G/\#G_B)$ Kandidaten testen ([CoF$^+$06], 20.3.1.).

In Schritt 3 des Algorithmus wird die lineare Abhängigkeit modulo den Primteilern von $\#G$ ermittelt. Normalerweise jedoch ist $\#G$ selbst bereits eine Primzahl. Die Matrix $A$ ist *sparse*, denn die Elemente von $G$ haben beschränkte Norm, also gibt es $O(\log_2 c) = O(\log_2 n'_B)$ von null verschiedene Einträge in $A$ für entsprechend großes $c$. Daher kann man Wiedemanns Algorithmus benutzen und erhält eine Laufzeit von $O(c^2 + c\omega)$, falls $\omega$ die Anzahl der von null verschiedenen Einträge von allen $e_{i,j}$ und $x_i$, $y_i$ ist oder $O(c^2)$, wenn man Lanczos Algorithmus benutzt.

Vernachlässigen wir die Zeit, die zum Faktorisieren von $\#G$ benötigt wird, ergibt sich eine Gesamtkomplexität von

$$\tilde{O}\left((n'_B)^2 + n'_B \frac{\#G}{\#G_B} \tau_s\right).$$

Dabei spielen die Glattheitsgrenze $B$ und die Komplexität $\tau_s$ eine wichtige Rolle bei der Bestimmung der Gesamtkomplexität eines Index Calculus Algorithmus

eines spezifischen Gruppentyps. Diese Überlegungen führen zu dem Ergebnis, dass der Index Calculus Algorithmus subexponentielle Laufzeit hat, wie der folgende Satz erläutert (s. [CoF$^+$06], Abschnitt 20.3.3.a).

**Satz 2.1.7** *Angenommen für $G$ kann die Glattheitsgrenze so gewählt werden, dass für $\rho, \sigma > 0$ gilt*

$$n'_B = L_{(\#G)}(1/2, \rho + o(1))$$

*und*

$$\frac{\#G}{\#G_B} = L_{(\#G)}(1/2, \sigma + o(1)).$$

*Weiterhin nehmen wir an, dass der Glattheitstest in $G$ in $\tilde{O}((n'_B)^\tau)$ für eine Konstante $\tau$ durchgeführt werden kann. Dann benötigen wir*

$$\tilde{O}\left(L_{(\#G_B)}\left(1/2, \max\{2\rho, 1 + (1+\tau)\rho + \sigma\} + o(1)\right)\right)$$

*Operationen in $G$ um das diskrete Logarithmusproblem in $G$ zu lösen.*

## 2.2 Punkezählalgorithmen auf Kurven

Nun betrachten wir Methoden für den speziellen Fall, dass die Gruppe $G$ die Punktegruppe der $\mathbb{F}_{p^n}$-rationalen Punkte der Jacobischen $J_C$ einer Kurve $C$ ist. Aufgrund der Spezialisierung der Gruppe $G$ hat man nun noch effizientere Methoden zur Verfügung. Diese basieren darauf, den Frobenius-Morphismus auf geeigneten Darstellungsräumen zu berechnen und das charakteristische Polynom des Frobenius auszuwerten.

Zu den speziellen Kurventypen gibt es eine Fülle von Algorithmen, so dass eine vollständige Darstellung aller Methoden den Rahmen dieser Arbeit übersteigen würde. Daher beschränken wir uns darauf, die Grundideen zu veranschaulichen, auf denen diese Algorithmen basieren und dann eine Auflistung der Komplexitäten einzelner Methoden anzugeben. Wir beginnen mit der Definition des Frobenius-Morphismus.

### 2.2.1 Der Frobenius-Morphismus

Für eine Einführung in die hier auftretenden Strukturen, siehe [Har77].

**Definition 2.2.1** *Sei $X$ ein Schema, dessen lokale Ringe alle Charakteristik $p > 0$ haben. Dann definieren wir den* ***absoluten Frobenius-Morphismus*** *$\phi_X : X \to X$ wie folgt: Sei $\phi_X$ die identische Abbildung auf dem topologischen Raum $|X|$ und $\phi_X^\sharp : \mathcal{O}_X \to \mathcal{O}_X$ die Abbildung, die jedes Element zur $p$-ten Potenz erhebt.*

Da die lokalen Ringe alle von Charakteristik $p$ sind, induziert $\phi_X^\sharp$ einen lokalen Homomorphismus auf den lokalen Ringen und daher ist $\phi_X$ ein Morphismus.

Ist $f : X \longrightarrow Y$ ein Morphismus zweier Schemata, deren lokale Ringe alle Charakteristik $p > 0$ haben und $\phi_X$ bzw. $\phi_Y$ der absolute Frobenius-Morphismus auf $X$ bzw. $Y$, dann haben wir das kommutative Diagramm

$$\begin{array}{ccc} X & \xrightarrow{\phi_X} & Y \\ \downarrow f & & \downarrow f \\ Y & \xrightarrow{\phi_Y} & Y. \end{array} \tag{2.4}$$

Wir betrachten nun die relative Situation, d.h. wir legen ein Schema $S$ zugrunde und betrachten Schemata über $S$. Ist $\alpha : X \longrightarrow S$ ein $S$-Schema, dann ist der absolute Frobenius-Morphismus $\phi_X$ im allgemeinen *kein* Morphismus von $S$-Schemata, außer z.B. bei $S = \mathrm{Spec}(\mathbb{F}_p)$. Um dem abzuhelfen, definieren wir $\alpha^{(p)} : X^{(p/S)} \to S$ als Pull-Back von $\alpha : X \to S$ via $\phi_S : S \to S$. Nach Definition ist $X^{(p/S)} = S \times_{\phi_S, S} X$ und wir haben das kartesische Diagramm

$$\begin{array}{ccc} X^{(p/S)} & \xrightarrow{h} & X \\ \downarrow \alpha^{(p)} & & \downarrow \alpha \\ S & \xrightarrow{\phi_S} & S. \end{array} \tag{2.5}$$

Wir schreiben vereinfachend $X^{(p)}$ für $X^{(p/S)}$, falls eine Verwechslung ausgeschlossen ist. Man beachte, dass dieses Schema im allgemeinen sehr von dem betrachteten Basis-Schema $S$ abhängt. Da das Diagramm (2.5) kartesisch ist, erhalten wir aus dem kommutativen Diagramm (2.4) durch Einsetzen von $Y = S$ ein kommutatives Diagramm

$$\begin{array}{ccccc} X & \xrightarrow{\phi_{X/S}} & X^{(p/S)} & \xrightarrow{W} & X \\ & & \downarrow \alpha^{(p)} & & \downarrow \alpha \\ & & S & \xrightarrow{\phi_S} & S. \end{array} \tag{2.6}$$

**Definition 2.2.2** *Der Morphismus von $S$-Schemata $\phi_{X/S} : X \to X^{(p/S)}$ wird* ***relativer Frobenius-Morphismus*** *von $X$ über $S$ genannt.*

Nach Definition ist $\phi_{X/S}$ ein Morphismus von $S$-Schemata (denn es gilt $\alpha^{(p)} \circ \phi_{X/S} = \alpha$) und $W \circ \phi_{X/S}$ ist der absolute Frobenius von $X$.

**Beispiel 2.2.3** Sei $S = \mathrm{Spec}(R)$ und $X = \mathrm{Spec}(R[t_1, \ldots, t_m]/I)$ für ein Ideal $I = (f_1, \ldots, f_n) \subset R[t_1, \ldots, t_m]$. Sei $f_i^{(p)} \in R[t_1, \ldots, t_m]$ das Polynom, das wir durch $p$-Potenzierung all seiner Koeffizienten (aber nicht der Variablen!) erhalten. Gilt also in Multi-Index Schreibweise $f_i = \sum c_\nu t^\nu$, dann ist $f^{(p)} = \sum c_\nu^p t^\nu$. Wir können $X^{(p)}$ darstellen, als $X^{(p)} = \mathrm{Spec}(R[t_1, \ldots, t_m]/I^{(p)})$ mit $I^{(p)} = (f_1^{(p)}, \ldots, f_n^{(p)})$ und der relative Frobenius-Morphismus $\phi_{X/S} : X \to X^{(p)}$ wird induziert durch den Homomorphismus

$$R[t_1, \ldots, t_m]/I^{(p)} \to R[t_1, \ldots, t_m]/I$$

mit $r \mapsto r$ für alle $r \in R$ und $t_j \mapsto t_j^p$. Dieser Homomorphismus ist wohldefiniert.

$\phi_{X/S} \circ W : X^{(p)} \to X^{(p)}$ ist gleich dem absoluten Frobenius-Morphismus auf $X^{(p)}$. Da der absolute Frobenius-Morphismus die Identität auf dem zugrundeliegenden topologischen Raum ist, folgt dass $\phi_{X/S} : X \to X^{(p)}$ einen Homeomorphismus auf den topologischen Räumen $|X| \xrightarrow{\sim} |X^{(p)}|$ induziert.

Die Formation des relativen Frobenius-Morphismus ist kompatibel mit Basiswechseln. Damit ist folgendes gemeint: Sei $\alpha : X \to S$ ein $S$-Schema und $T \to S$ ein weiters Schema über $S$. Betrachte den Morphismus $\alpha_T : X_T \to T$, den wir aus $\alpha$ durch Basiswechsel erhalten. Die erste Beobachtung ist, dass $(X_T)^{(p/T)}$ kanonisch Isomorph zu $(X^{(p/S)})_T$ ist. Identifizieren wir diese beiden Schemata, dann ist der relative Frobenius $\phi_{X_T/T}$ von $X_T$ über $T$ gleich dem Pull-Back $(\phi_{X/S})_T$ des relativen Frobenius von $X$ über $S$.

Der absolute und der relative Frobenius-Morphismus können iteriert werden. Dies ergibt sich für den absoluten Frobenius-Morphismus unmittelbar: $\phi_X^n : X \to X$ ist einfach die $n$-fache Anwendung von $\phi_X$. Die $n$-fache Anwendung des relativen Frobenius-Morphismus ist ein Morphismus $\phi_{X/S}^n : X \to X^{(p^n/S)}$. Diese Definition ist eine einfache Generalisierung der Definition von $\phi_{X/S}$: wir definieren $\alpha^{(p^n)} : X^{(p^n/S)} \to S$ als Pull-Back von $\alpha : X \to S$ via $\phi_S^n$. Dann faktorisiert $\phi_X^n$ als

$$X \xrightarrow{\phi_{X/S}^n} X^{(p^n/S)} \xrightarrow{h^{(n)}} X$$

mit $\alpha^{(p^n)} \circ \phi_{X/S}^n = \alpha$. Anders ausgedrückt:

$$X^{(p^2/S)} = \left(X^{(p/S)}\right)^{(p/S)}, \quad X^{(p^3/S)} = \left(X^{(p^2/S)}\right)^{(p/S)}, \quad \text{usw.},$$

und

$$\phi_{X/S}^n = \left( X \xrightarrow{\phi_{X/S}} X^{(p)} \xrightarrow{\phi_{X^{(p)}/S}} X^{(p^2)} \longrightarrow \cdots \xrightarrow{\phi_{X^{(p^{n-1})}/S}} X^{(p^n)} \right).$$

Sei $S = \mathrm{Spec}(\mathbb{F}_q)$ mit $q = p^n$. Ist $X$ ein $S$-Schema, dann ist die $n$-fache Anwendung des absoluten Frobenius-Morphismus $\phi_X^n : X \to X$ ein Morphismus von $S$-Schemata. Es gilt $\phi_X^n = \phi_{X/S}^n$.

**Definition 2.2.4** *Für ein* $\mathrm{Spec}(\mathbb{F}_q)$*-Schema* $X$ *bezeichnen wir* $\phi_q := \phi_X^n$ *als* ***geometrischen Frobenius-Morphismus*** *von* $X$*.*

Sei $\mathbb{F}_q$ ein endlicher Körper mit $q = p^d$ für eine Primzahl $p$. Für eine projektive nichtsinguläre Kurve $C$ von Geschlecht $g$ über $\mathbb{F}_q$ sei $\phi_q$ der Frobenius-Morphismus von $J_C$. Dann sind die unter $\phi_q$ invarianten Elemente genau die Menge $J_C(\mathbb{F}_q) = \ker(\phi_q - [1])$. Wir fassen einige wichtige Resultate in dem folgenden Satz zusammen.

**Satz 2.2.5** *$\phi_q$ ist ein Endomorphismus von $J_C$. Das charakteristische Polynom $\chi_{\phi_q}(T)$ von $\phi_q$ in Abhängigkeit von $C$ ist ein normiertes Polynom der Form*

$$\chi_{\phi_q}(T) = T^{2g} + a_1 T^{2g-1} + \cdots + a_g T^g + \cdots + a_1 q^{g-1} T + q^g$$

*mit Koeffizienten $a_i$ in $\mathbb{Z}$. Es gilt*

$$\#J_C(\mathbb{F}_q) = \chi_{\phi_q}(1).$$

### 2.2.2 Die Zeta-Funktion und der Satz von Weil

Wir führen nun die Zeta-Funktion einer glatten projektiven Kurve ein und beschreiben ihren Zusammenhang mit $\chi_{\phi_q}(T)$.

**Definition 2.2.6** *Sei $N_k$ die Anzahl der $\mathbb{F}_{q^k}$-rationalen Punkte von $C$. Die **Zeta-Funktion** $Z(C/\mathbb{F}_q;T)$ von $C$ über $\mathbb{F}_q$ ist die Erzeugendenfunktion*

$$Z(C/\mathbb{F}_q;T) = \exp\left(\sum_{k=1}^{\infty} \frac{N_k}{k} T^k\right).$$

Die Zeta-Funktion sollte als formale Potenzreihe mit Koeffizienten in $\mathbb{Q}$ betrachtet werden. 1949 beschrieb Weil den folgenden Satz als Vermutungen für glatte projektive Varietäten, die Deligne 1972 vollständig bewies.

**Satz 2.2.7 (Weil, 1941)** *Sei $C$ eine glatte projektive Kurve von Geschlecht $g$ über einem endlichen Körper mit $q$ Elementen.*

1. ***Rationalität:** $Z(C/\mathbb{F}_q;T) \in \mathbb{Q}[[T]]$ ist eine rationale Funktion.*
2. ***Funktionalgleichung:** $Z(T) = Z(C/\mathbb{F}_q;T)$ erfüllt*

$$Z\left(\frac{1}{qT}\right) = \pm q^{1-g} T^{2-2g} Z(T).$$

3. ***Riemann-Vermutung:** Definiere das **$L$-Polynom** von $C$ durch*

$$L(T) = T^{2g} \chi_{\phi_q}\left(\frac{1}{T}\right).$$

*Dann ist die Zeta-Funktion von $C$ gegeben durch*

$$Z(C/\mathbb{F}_q;T) = \frac{L(T)}{(1-T)(1-qT)}.$$

*Schreiben wir $L(T) = \prod_{i=1}^{2g}(1-\alpha_i T)$, dann folgt $|a_i| = \sqrt{q}$.*

Sei $L(T) = a_0 + a_1 T + \cdots + a_{2g} T^{2g}$, dann zeigt die Funktionalgleichung, dass $a_{2g-i} = q^{g-i} a_i$ für $i = 0, \ldots, g$. Weiterhin können wir nach der Funktionalgleichung die $\alpha_i$ so bezeichnen, dass gilt $\alpha_i \alpha_{i+g} = q$ für $i = 0, \ldots, g$. Logarithmieren wir beide Darstellungen der Zeta-Funktion, so erhalten wir

$$\ln\, Z(C/\mathbb{F}_q;T) = \sum_{k=1}^{\infty} \frac{N_k}{k} T^k = \sum_{i=1}^{2g} \ln(1-\alpha_i T) - \ln(1-T) - \ln(1-qT).$$

Wegen $\ln(1-sT) = -\sum_{i=1}^{\infty} \frac{(sT)^k}{k} T^k$ schliessen wir für alle positiven $k$, dass gilt

$$N_k = q^k + 1 - \sum_{i=1}^{2g} \alpha_i^k.$$

### 2.2.3 Kohomologie und der Fixpunktsatz von Lefschetz

In diesem Abschnitt veranschaulichen wir, wie man den Satz 2.2.7 von Weil aus Kohomologietheorie ableiten kann. Sei $C$ wieder eine glatte projektive Kurve von Geschlecht $g$ über einem endlichen Körper $\mathbb{F}_q$ der Charakteristik $p$ und sei $\overline{C} \otimes_{\mathbb{F}_q} \overline{\mathbb{F}}_q$ die korrespondierende Kurve über dem algebraischen Abschluss $\overline{\mathbb{F}}_q$ von $\mathbb{F}_q$.

Sei $l$ eine von $p$ verschiedene Primzahl und $\mathbb{Q}_l$ der Körper der $l$-adischen Zahlen und bezeichne $H^i(\overline{C}, \mathbb{Q}_l)$ die $l$-adischen Kohomologiegruppen. Auf eine Beschreibung dieser Gruppen können wir hier aus Platzgründen nicht eingehen und beschränken uns auf die Angabe ihrer Haupteigenschaften, siehe [SGA4] für Details. Da $C$ jedoch eine glatte projektive Kurve ist, haben wir den folgenden Satz.

**Satz 2.2.8** *Sei $C$ eine glatte projektive Kurve über einem endlichen Körper $\mathbb{F}_q$ der Charakteristik $p$ und sei $l$ eine von $p$ verschiedene Primzahl. Dann existiert ein Isomorphismus*

$$H^1(C, \mathbb{Z}_l) \simeq T_l(J_C).$$

*Dabei definieren wir den* ***$l$-adischen Tate-Modul*** *als*

$$T_l(J_C) := \lim_{\leftarrow} J_C[l^n],$$

*mit den $l^n$-Torsionspunktegruppen $J_C[l^n]$ von $J_C$.*

Um die Rationalität der Zeta-Funktion und die Zerlegung ihres Zählers und Nenners zu zeigen, benötigen wir nur die folgenden Eigenschaften:

- Die $l$-adischen Kohomologiegruppen $H^i(\overline{C}, \mathbb{Q}_l)$ sind endlich dimensionale Vektorräume über $\mathbb{Q}_l$ und $H^i(\overline{C}, \mathbb{Q}_l) = 0$ für $i < 0$ und $i > 2$.
- Sei $f : \overline{C} \to \overline{C}$ ein Morphismus mit isolierten Fixpunkten und jeder dieser Fixpunkte habe Multiplizität 1. Dann ist die Zahl $N(f, C)$ der Fixpunkte von $f$ durch den *Fixpunktsatz von Lefschetz* gegeben:

$$N(f, C) = \sum_{i=0}^{2} (-1)^i \mathrm{Tr}(f^*; H^i(\overline{C}, \mathbb{Q}_l)).$$

Wir wissen, dass die Zahl $N_k$ von $\mathbb{F}_q$-rationalen Punkten von $C$ gleich der Anzahl der Fixpunkte von $\phi_q^k$ des Frobenius-Endomorphismus $\phi_q$ ist. Der Fixpunktsatz von Lefschetz impliziert

$$N_k = \sum_{i=0}^{2} (-1)^i \mathrm{Tr}(\phi_q^{k*}; H^i(\overline{C}, \mathbb{Q}_l)).$$

Setzen wir dies in die Definition der Zeta-Funktion ein, so erhalten wir

**Satz 2.2.9** *Sei $C$ eine glatte projektive Kurve von Geschlecht $g$ über $\mathbb{F}_q$, dann gilt*

$$Z(C/\mathbb{F}_q;T) = \frac{L(T)}{(1-T)(1-qT)}$$

*mit*

$$L(T) = \det\left(1 - \phi_q^* T; H^1(\overline{C}, \mathbb{Q}_l)\right).$$

Der obige Satz begründet den ersten kohomologischen Ansatz für die Berechnung der Zeta-Funktion einer glatten projektiven algebraischen Varietät: konstruiere eine Basis der $l$-adischen Kohomologiegruppe $H^1(\overline{C}, \mathbb{Q}_l)$. Leider ist die Definition von $H^1(\overline{C}, \mathbb{Q}_l)$ sehr abstrakt und ermöglicht keine algorithmischen Berechnungsmöglichkeiten. Nach Satz 2.2.8 haben wir jedoch den Isomorphismus $H^1(C, \mathbb{Q}_l) \simeq T_l(J_C)$.

Der zweite kohomologische Ansatz konstruiert $p$-adische Kohomologiegruppen über einer unverzweigten Erweiterung $\mathbb{Q}_q$ von $\mathbb{Q}_p$. Es gibt viele verschiedene Theorien, in denen ein Fixpunktsatz von Lefschetz existiert, z.B. Monsky-Washnitzer Kohomologie, Lubkin's $p$-adische Kohomologie, kristallinie Kohomologie nach Grothendieck und Berthelot und letztlich rigide Kohomologie nach Berthelot, siehe [CoF$^+$06], Abschnitt 8.1.2 für weiterführende Literatur. Der größte Vorteil dieser Algorithmen gegenüber $l$-adischer Kohomologietheorie ist die Existenz von Isomorphismen mit algebraischer de Rham Kohomologie, d.h. Differentialen modulo exakten Differentialen. Die algebraische de Rham Kohomologie selbst ist effizient zu berechnen und daher für Berechnungen zugänglicher als die $l$-adische Kohomologie. Der Vorteil der $l$-adischen Methode gegenüber der $p$-adischen Vorgehensweise ist, dass man bei der $l$-adischen Methode die Informationen aus mehreren kleinen Zahlen $l$ kombinieren kann.

## 2.3 $l$-adische Methoden

Sei $\mathbb{F}_q$ ein endlicher Körper der Charakteristik $p$ und sei $C$ eine glatte projektive Kurve von Geschlecht $g$ über $\mathbb{F}_q$. Die jacobische Varietät $J_C$ von $C$ über $\mathbb{F}_q$ ist eine abelsche Varietät der Dimension $g$. Sei $\chi_{\phi_q}(T) \in \mathbb{Z}[T]$ das charakteristische Polynom des Frobenius-Endomorphismus auf dem Tate-Modul $T_l(J_C)$ für eine von $p$ verschiedene Primzahl $l$. Dann können wir schreiben

$$\chi_{\phi_p}(T) = \sum_{i=0}^{2g} a_{2g-i} T^i \text{ mit } a_0 = 1 \text{ und } a_{2g} = q^g.$$

Nach der Funktionalgleichung der Zeta-Funktion haben wir $a_{2g-i} = q^{g-1} a_i$ für $i = 0, \ldots, g$, also reicht es, $a_i$ für $i = 0, \ldots, g$ zu bestimmen.

Die Grundidee der $l$-adischen Methoden ist, $T_l(J_C)$ durch die $l$-Torsionspunkte $J_C[l]$ zu approximieren. Da $l \neq p$ ist die $l$-Torsion ein $2g$-dimensionaler Vektorraum über $\mathbb{Z}/l\mathbb{Z}$ und die Einschränkung von $\phi_q$ auf $J_C[l]$ ist eine lineare Transformation dieses Vektorraums. Sei $P_l(T)$ das charakteristische Polynom dieser Einschränkung, dann gilt (s. [CoF$^+$06], Satz 5.71)

$$P_l(T) \equiv \chi_{\phi_q}(T) \bmod l. \tag{2.7}$$

Nach der Riemann-Vermutung gilt weiterhin, dass die Koeffizienten $a_0, \ldots, a_g$ eingeschränkt sind durch

$$|a_i| \leq \left( \begin{array}{c} 2g \\ i \end{array} \right) q^{i/2} \leq \left( \begin{array}{c} 2g \\ g \end{array} \right) q^{g/2}$$

Mit Hilfe des chinesischen Restsatzes können wir also $\chi_{\phi_q}$ eindeutig aus $P_l(T)$ für Primzahlen $l \leq H \ln(q)$ bestimmen. $H$ ist dabei eine Konstante, für die gilt

$$\prod_{\substack{\text{Primzahlen } l \ \leq \ H \ln(q) \\ \text{ggT}(l,q)=1}} l \quad > \quad 2 \left( \begin{array}{c} 2g \\ g \end{array} \right) q^{g/2}.$$

Die Konstante $H$ hängt nur von $g$ ab und der Primzahlsatz impliziert, dass $H$ linear in $g$ ist.

Für vorgegebenes $l$ kann das Polynom $P_l$ folgendermassen berechnet werden. Angenommen $J_C$ ist eingebettet in $\mathbb{P}^N$ und ein affiner Teil wird definiert durch die Polynomgleichungen $F_1, \ldots, F_s \in \mathbb{F}_q[X]$ mit $X = (X_1, \ldots, X_N)$. Weiterhin nehmen wir an, dass das Additionsgesetz explizit durch ein $N$-Tupel rationaler Funktionen $(G_1(X,Y), \ldots, G_N(X,Y))$ mit $Y = (Y_1, \ldots, Y_N)$ gegeben ist. Mit der Verdoppeln-und-Addieren-Methode berechnen wir die Menge der Polynome $Q_1^l, \ldots, Q_{k_l}^l$ des Ideals der Untervarietät der $l$-Torsionspunkte von $J_C$. Sei $I_l$ das Radikalideal von $\langle F_1, \ldots, F_s, Q_1^l, \ldots, Q_{k_l}^l \rangle$. Um $P_l$ zu bestimmen, suchen wir ganze Zahlen $0 \leq a_i < l$ mit $i = 0, \ldots, 2g$, so dass

$$\sum_{i=0}^{2g} [a_{2g-i}](X_1^{q^i}, \ldots X_N^{q^i}) \in I_l,$$

wobei die Addition in der obigen Gleichung das Gruppengesetz auf $J_C$ und $[m]$ die Multiplikation mit $m$ in $J_C$ darstellt. Der resultierende Algorithmus hat Komplexität $O(\lg(q)^\Delta)$, wobei $\Delta$ nur von dem Geschlecht $g$, der Dimension des Einbettungsraums $\mathbb{P}^N$, der Anzahl und den Graden der definierenden Gleichungen für $J_C$ un dem Gruppengesetz abhängt. Siehe [Pil90] für weitere Details.

**Bemerkung 2.3.1** Obwohl der obige Algorithmus eine polynomielle Zeitkomplexität in $\lg(q)$ hat, ist er derzeit nur praktikabel für elliptische Kurven und hyperelliptische Kurven von Geschlecht 2. Der Grund dafür ist, dass die Polynome $Q_1^l, \ldots, Q_{k_l}^l$ wie $O(l^{2g})$ wachsen, da $J_C[l] \simeq (\mathbb{Z}/l\mathbb{Z})^{2g}$. Für elliptische Kurven kann der Algorithmus jedoch wesentlich verbessert werden, indem man die Einschänkung auf eine Untergruppe von $J_C[l]$ betrachtet, welche der Kern einer Isogenie von Grad $l$ ist, siehe [CoF$^+$06], Abschnitt 17.2.

## 2.4 p-adische Methoden

Die bekannteste Anwendung $p$-adischer Methoden in der algebraischen Geometrie ist sicherlich Dworks genialer Beweis der Rationalität der Zeta-Funktion

von 1960. Obwohl DWORKs Beweis leicht in einen Algorithmus zur Berechnung der Zeta-Funktion einer beliebigen algebraischen Varietät umgewandelt werden kann, wurde dies von niemandem erkannt und für mehr als zehn Jahre wurden ausschliesslich $l$-adische Algorithmen genutzt. 1999 führte SATOH den $p$-adischen Ansatz in die rechnergestützte algebraische Geometrie ein, indem er einen $p$-adischen Algorithmus zur Berechnung der Punkteanzahl einer gewöhnlichen elliptischen Kurve über einem endlichen Körper durch $p$-adische Liftung vorstellte. Diesem Durchbruch folgten viele $p$-adische Theorien, die als Basis für neue Algorithmen verwendet wurden. In diesem Abschnitt stellen wir die zwei für praktische Berechnungen wichtigsten Theorien vor, dies sind die Serre-Tate-Liftung und die Monsky-Washnitzer Kohomologie.

### 2.4.1 Die kanonische Liftung nach Serre-Tate

Sei $\overline{\mathcal{A}}$ eine abelsche Varietät über $\mathbb{F}_q$ mit $q = p^d$ und $p$ prim. Sei $\mathbb{Q}_q$ eine unverzweigte Erweiterung von Grad $d$ von $\mathbb{Q}_p$ mit Bewertungsring $\mathbb{Z}_q$ und Restklassenkörper $\mathbb{Z}_q/(p\mathbb{Z}_q) \simeq \mathbb{F}_q$. Wir betrachten eine beliebige Liftung $\mathcal{A}$ von $\overline{\mathcal{A}}$ über $\mathbb{Z}_q$ das heißt, die Reduktion von $\mathcal{A}$ modulo $p$ ist $\overline{\mathcal{A}}$. Im allgemeinen gibt es dann keinen Endomorphismus $\mathcal{F} \in \mathrm{End}(\mathcal{A})$, der unter dieser Reduktion zu dem $q$-Potenz Frobenius Endomorphismus $\phi_q \in \mathrm{End}(\overline{\mathcal{A}})$ wird.

**Definition 2.4.1** *Eine **kanonische Liftung** einer abelschen Varietät $\overline{\mathcal{A}}$ über $\mathbb{F}_q$ ist eine abelsche Varietät $\mathcal{A}$ über $\mathbb{Q}_q$, so dass $\mathcal{A}$ modulo $p$ zu $\overline{\mathcal{A}}$ reduziert und der durch Reduktion modulo $p$ induzierte Ringhomomorphismus* $\mathrm{End}(\mathcal{A}) \to \mathrm{End}(\overline{\mathcal{A}})$ *ein Isomorphismus ist.*

Falls $\overline{\mathcal{A}}$ eine kanonische Liftung $\mathcal{A}_c$ zulässt, impliziert diese Definition, dass es eine Liftung $\mathcal{F} \in \mathrm{End}(\mathcal{A}_c)$ des Frobenius Endomorphismus $\phi_q \in \mathrm{End}(\overline{\mathcal{A}})$ gibt. Tatsächlich gilt sogar die Umkehrung: Sei $\mathcal{A}$ eine Liftung von $\overline{\mathcal{A}}$ und $\mathcal{F} \in \mathrm{End}(\mathcal{A})$ reduziert zu $\phi_q \in \mathrm{End}(\overline{\mathcal{A}})$, dann ist $\mathcal{A}$ eine kanonische Liftung von $\overline{\mathcal{A}}$. 1941 bewies DEURING, dass eine gewöhnliche elliptische Kurve immer eine kanonische Liftung existiert und diese bis auf Isomorphie eindeutig ist. 1967 verallgemeinerten LUBIN, SERRE und TATE diese Aussage auf allgemeine abelsche Varietäten.

**Satz 2.4.2 (Lubin-Serre-Tate)** *Sei $\mathcal{A}$ eine **gewöhnliche** abelsche Varietät über $\mathbb{F}_q$. Dann gibt es eine kanonische Liftung $\mathcal{A}_c$ von $\overline{\mathcal{A}}$ über $\mathbb{Z}_q$ und $\mathcal{A}_c$ ist eindeutig bestimmt bis auf Isomorphie.*

Eine abelsche Varietät $\overline{\mathcal{A}}$ ist gewöhnlich, falls sie maximalen $p$-Rang hat, das heißt $\overline{\mathcal{A}}[p] = (\mathbb{Z}/p\mathbb{Z})^{\dim(\overline{\mathcal{A}})}$. Zur Konstruktion einer $p$-adischen Approximation von $\mathcal{A}_c$ zu gegebenem $\overline{\mathcal{A}}$ verfährt man wie folgt: Sei $\mathcal{A}_0$ die Liftung von $\overline{\mathcal{A}}$ über $\mathbb{Z}_q$ und bezeichne mit $\pi : \mathcal{A}_0 \to \overline{\mathcal{A}}$ die Reduktion modulo $p$. Betrachte die Untergruppe $\mathcal{A}_0[p]^{\mathrm{loc}} = \mathcal{A}_0[p] \cap \ker(\pi)$, also die $p$-Torsionspunkte von $\mathcal{A}_0$, die auf das neutrale Element von $\overline{\mathcal{A}}$ reduzieren. Nach einem Satz von CARLS gilt dann, dass $\mathcal{A}_1 = \mathcal{A}_0/\mathcal{A}_0[p]^{\mathrm{loc}}$ wieder eine abelsche Varietät, so dass ihre Reduktion gewöhnlich ist und es exisitiert eine Isogenie $I_0 : \mathcal{A}_0 \to \mathcal{A}_1$, die auf den $p$-Potenz Frobenius Endomorphismus $\sigma : \overline{\mathcal{A}} \to \overline{\mathcal{A}}$ reduziert. Rekursiv definieren wir also

$\mathcal{A}_i = \mathcal{A}_{i-1}/\mathcal{A}_{i-1}[p]^{\text{loc}}$ für positives $i$ und erhalten eine Sequenz von abelschen Varietäten und Isogenien

$$\mathcal{A}_0 \xrightarrow{I_0} \mathcal{A}_1 \xrightarrow{I_1} \mathcal{A}_2 \xrightarrow{I_2} \mathcal{A}_3 \xrightarrow{I_3} \dots$$

Offensichtlich reduziert $\mathcal{A}_{kd}$ mit $k \in \mathbb{N}$ zu $\overline{\mathcal{A}}$ modulo $p$, außerdem konvergiert die Folge $\{\mathcal{A}_{kd}\}_{k\in\mathbb{N}}$ auf die kanonische Liftung $\mathcal{A}_c$ und die Konvergenz ist linear.

Sei $C$ eine glatte projektive Kurve von Geschlecht $g$ über $\mathbb{F}_q$ mit jacobischer Varietät $J_C$. Ist $J_C$ gewöhnlich, dann können wir ihre Liftung $\mathcal{A}_c$ betrachten. Man beachte, dass $\mathcal{A}_c$ dabei nicht selbst die Jacobische einer Kurve sein muss. Da $\text{End}(\mathcal{A}_c)$ isomorph zu $\text{End}(J_C)$ ist, existiert eine Liftung $\mathcal{F}$ des Frobenius-Endomorphismus $\phi_q$.

Um das charakteristische Polynom von $\phi_q$ zu berechnen, gehen wir wie folgt vor: Sei $D_0(\mathcal{A}_c, \mathbb{Q}_q)$ der Raum der holomorphen Differentialformen von Grad 1 auf $\mathcal{A}_c$ über $\mathbb{Q}_q$, dann gilt $\dim(D_0(\mathcal{A}_c, \mathbb{Q}_q)) = g$, wegen $\dim(J_C) = g$. Wählen wir eine Basis $B$ von $D_0(\mathcal{A}_c, \mathbb{Q}_q)$, dann kann jeder Endomorphismus $\lambda \in \text{End}_{\mathbb{Q}_q}(\mathcal{A}_c)$ durch eine $g \times g$ Matrix $M$ über $\mathbb{Q}_q$ dargestellt werden, indem man die Operation von $\lambda^*$ auf $B$ betrachtet, also $\lambda^* = MB$. Die Verbindung zu dem charakteristischen Polynom $\chi_{\phi_q}$ des Frobenius wird durch den folgenden Satz hergestellt.

**Satz 2.4.3** *Sei $\mathcal{F} \in \text{End}_{\mathbb{Q}_q}(\mathcal{A}_c)$ die Liftung des Frobenius-Endomorphismus $\phi_q \in \text{End}(J_C)$ und $M_{\mathcal{F}}$ die Matrix, die die Operation von $\phi_q^*$ auf $D_0(\mathcal{A}_c, \mathbb{Q}_q)$ repräsentiert. Ist $P(T) \in \mathbb{Z}_q(T)$ das charakteristische Polynom von $M_{\mathcal{F}} + qM_{\mathcal{F}}^{-1}$, dann wird das charakteristische Polynom $\chi_{\phi_q}$ gegeben durch*

$$\chi\phi_q = T^g P\left(T + \frac{q}{T}\right).$$

Beachte, dass wir auch $\chi_{\phi_q}(T) = P_1(T)P_2(T)$ schreiben können, wobei $P_1$ das charakteristische Polynom von $M_{\mathcal{F}}$ und $P_2$ das charakteristische Polynom von $qM_{\mathcal{F}}^{-1}$ ist.

Die Punktezählungsalgorithmen basierend auf der kanonischen Liftung verlaufen also in zwei Schritten: Im ersten Schritt wird eine ausreichend genaue Approximation der kanonischen Liftung von $J_C$ (oder ihrer Invarianten) erstellt und im zweiten Schritt wird die Operation der Liftung $\mathcal{F}$ des Frobenius-Endomorphismus auf $D_0(\mathcal{A}_c, \mathbb{Q}_q)$ berechnet.

### 2.4.2 Monsky-Washnitzer Kohomologie

In diesem Abschnitt spezialisieren wir den Formalismus der Monsky-Washnitzer Kohomologie auf affine ebene Kurven. Sei $\overline{\mathcal{C}}$ eine affine ebene Kurve über einem Körper $\mathbb{F}_q$ mit $q = p^d$ Elementen und sei $\mathbb{Q}_q$ eine unverzweigte Erweiterung von $\mathbb{Q}_p$ von Grad $d$ mit Bewertungsring $\mathbb{Z}_q$, so dass $\mathbb{Z}_q/p\mathbb{Z}_q = \mathbb{F}_q$. Das Ziel der Monsky-Washnitzer Kohomologie ist es, die Zeta-Funktion der Kurve $\overline{\mathcal{C}}$ durch den Frobenius Operator $\mathcal{F}$ zu beschreiben, der auf den $p$-adischen Kohomologiegruppen $H^i(\overline{\mathcal{C}}, \mathbb{Q}_q)$ von $\overline{\mathcal{C}}$ über $\mathbb{Q}_q$ wirkt. Man beachte, dass hier mit hoher $p$-adischer Approximation gearbeitet werden muss, da wir sonst nur die Zeta-Funktion modulo $p$ erhalten würden. Zu diesem Fall kommen wir im nächsten

Abschnitt. Für glatte Kurven sind die meisten dieser Gruppen trivial, wie der folgende Satz zeigt.

**Satz 2.4.4** *Sei $\overline{\mathcal{C}}$ eine glatte affine Kurve über dem endlichen Körper $\mathbb{F}_q$, dann sind die einzigen nichttrivialen Monsky-Washnitzer Kohomologiegruppen die Gruppen $H^0(\overline{\mathcal{C}}, \mathbb{Q}_q)$ und $H^1(\overline{\mathcal{C}}, \mathbb{Q}_q)$.*

Im weiteren Verlauf dieses Abschnitts wollen wir die Kohomologiegruppen $H^0(\overline{\mathcal{C}}, \mathbb{Q}_q)$ und $H^1(\overline{\mathcal{C}}, \mathbb{Q}_q)$ einführen und ihre wichtigsten Eigenschaften vorstellen.

Da $\overline{\mathcal{C}}$ eine ebene Kurve ist, kann $\overline{\mathcal{C}}$ durch ein biviarate Polynomgleichung $\overline{g}(x, y) = 0$ mit $\overline{g} \in \mathbb{F}_q[x, y]$ gegeben werden. Sei $\overline{A} = \mathbb{F}_q[x, y]/(\overline{g}(x, y))$ der Koordinatenring von $\overline{\mathcal{C}}$. Betrachte eine beliebige Liftung $g(x, y) \in \mathbb{Z}_q[x, y]$ von $\overline{g}(x, y)$ und sei $\mathcal{C}$ die durch $g(x, y) = 0$ beschriebene Kurve mit Koordinatenring $A = \mathbb{Z}_q[x, y]/(g(x, y))$. Um die Zeta-Funktion von $\overline{\mathcal{C}}$ mit Hilfe des Frobenius Operators zu beschreiben, benötigen wir eine Liftung des Frobenius-Endomorphismus $\phi_q$ auf $\overline{A}$ auf einen Operator auf der $\mathbb{Z}_q$-Algebra $A$. Wie wir in dem vorhergehenden Abschnitt gesehen haben, ist dies jedoch fast niemals möglich. Außerdem hängt die $\mathbb{Z}_q$-Algebra $A$ stark von der Wahl der Liftung ab, wie die folgenden Beispiele verdeutlichen.

**Beispiel 2.4.5** Sei $\overline{\mathcal{C}} : xy - 1 = 0$ über $\mathbb{F}_p$ mit Koordinatenring $\overline{A} = \mathbb{F}_p[x, 1/x]$. Wir betrachten die zwei Liftungen

$$g_1(x, y) = xy - 1 \qquad g_2(x, y) = x(1 + px)y - 1$$

und erhalten $A_1 = \mathbb{Z}_p[x, 1/x]$ und $A_2 = [x, 1/(x(1+px))]$, welche nicht isomorph zueinander sind.

Eine erste Abhilfe für beide Schwierigkeiten ist die Zuhilfenahme der $p$-adischen Komplettierung $A^\infty$ von $A$, welche bis auf Isomorphie eindeutig ist und eine Liftung von $\phi_q$ auf $A^\infty$ erlaubt. Dann allerdings entsteht ein neues Problem, denn die de Rham Kohomologie von $A^\infty$, auf die unser Vektorraum aufbaut, ist zu groß.

**Beispiel 2.4.6** Betrachte die affine Gerade über $\mathbb{F}_q$. Dann gilt $A = \mathbb{Z}_q[x]$ und $A^\infty$ ist der Potenzreihenring

$$\sum_{i=0}^{\infty} r_i x^i \quad \text{mit } r_i \in \mathbb{Z}_q \text{ und } \lim_{i \to \infty} r_i = 0.$$

Wir würden gerne $H^1(\overline{A}, \mathbb{Q}_q)$ definieren, als $A^\infty dx / d(A^\infty) \otimes_{Z_q} \mathbb{Q}_q$, aber dies ist unendlich-dimensional. Zum Beispiel ist jeder Term in der Differentialform $\sum_{i=0}^{\infty} p^i x^{p^i - 1} dx$ exakt, aber dessen Summe nicht, da $\sum_{i=0}^{\infty} x^{p^i}$ nicht in $A^\infty$ enthalten ist. Das fundamentale Problem ist, dass $\sum_{i=0}^{\infty} p^i x^{p^i - 1}$ nicht schnell genug konvergiert, damit dessen Integral ebenfalls konvergiert.

Monsky und Washnitzer arbeiteten daher mit einer Unteralgebra $A^\dagger$ von $A^\infty$, deren Elemente passende Wachstumsbedingungen erfüllen.

**Definition 2.4.7** *Sei $A = \mathbb{Z}_q[x,y]/(g(x,y))$, dann wird die **schwache Komplettierung** $A^\dagger$ definiert als $A^\dagger := Z_q\langle x,y\rangle^\dagger/(g(x,y))$, dabei ist $\mathbb{Z}_q\langle x,y\rangle^\dagger$ der Ring der überkonvergenten Potenzreihen*

$$\left\{\sum r_{i,j}x^iy^j \in \mathbb{Z}[[x,y]] \mid \exists\delta,\varepsilon\in\mathbb{R},\varepsilon>0,\forall(i,j): v_p(r_{i,j})\geq\varepsilon(i+j)+\delta\right\}.$$

Der Ring $A^\dagger$ erfüllt $A^\dagger/(pA^\dagger) = \overline{A}$ und hängt bis auf $\mathbb{Z}_q$-Isomorphie nur von $\overline{A}$ ab. Zudem haben Monsky und Washnitzer gezeigt, dass für jeden $\mathbb{F}_q$-Isomorphismus $\overline{\varphi}$ von $\overline{A}$ ein $\mathbb{Z}_q$-Endomorphismus $\varphi$ von $A^\dagger$ existiert, der eine Liftung von $\overline{\varphi}$ ist.

Zu jedem Element $s \in A^\dagger$ können wir das Differential $ds$ assoziieren, so dass die bekannte Leibniz-Regel gilt: für $s,t \in A^\dagger : d(st) = sdt + tds$. Dies impliziert $d(a) = 0$ für $a \in \mathbb{Z}_q$. Die Menge dieser Differentiale ist offensichtlich ein Modul über $A^\dagger$ und wird mit $D^1(A^\dagger)$ bezeichnet. Das folgende Lemma gibt eine präzise Beschreibung dieses Moduls.

**Lemma 2.4.8** *Der Universalmodul $D^1(A^\dagger)$ der Differentiale erfüllt*

$$D^1(A^\dagger) = (A^\dagger dx + A^\dagger dy)/(A^\dagger(g_x dx + g_y dy)),$$

*wobei $g_x = \frac{\partial g}{\partial x}$ und $g_y = \frac{\partial g}{\partial y}$ die partiellen Ableitungen von $g$ sind.*

Betrachten wir die totale Ableitung der Gleichung $g(x,y) = 0$, dann erhalten wir $g_x dx + g_y dy = 0$, was den Modul $A^\dagger(g_x dx + g_y dy)$ im obigen Lemma rechtfertigt. Die Abbildung $d : A^\dagger \longrightarrow D^1(A^\dagger)$ ist eine wohldefinierte Derivation, also ist es sinnvoll, ihren Kern und Kokern zu betrachten.

**Definition 2.4.9** *Die Kohomologiegruppen $H^0(\overline{A},\mathbb{Q}_q)$ und $H^1(\overline{A},\mathbb{Q}_q)$ sind definiert durch*

$$H^0(\overline{A},\mathbb{Q}_q) = \ker(d)\otimes_{\mathbb{Z}_q}\mathbb{Q}_q \quad \textit{und} \quad H^1(\overline{A},\mathbb{Q}_q) = \operatorname{coker}(d)\otimes_{\mathbb{Z}_q}\mathbb{Q}_q.$$

Nach Definition gilt $H^1(\overline{A},\mathbb{Q}_q) = \left(D^1(A^\dagger)/d(A^\dagger)\right)\otimes_{\mathbb{Z}_q}\mathbb{Q}_q$. Die Elemente von $d(A^\dagger)$ werden *exakte Differentiale* genannt. Man kann zeigen, dass $H^0(\overline{A},\mathbb{Q}_q)$ und $H^1(\overline{A},\mathbb{Q}_q)$ wohldefinierte, endlich dimensionale Vektorräume über $\mathbb{Q}_q$ sind, die nur von $\overline{A}$ abhängen.

**Satz 2.4.10** *Sei $\overline{\mathcal{C}}$ eine glatte affine Kurve von Geschlecht $g$, dann ist $\dim(H^0(\overline{A},\mathbb{Q}_q)) = 1$ und $H^1(\overline{A},\mathbb{Q}_q) = 2g+m-1$, wobei $m$ die Anzahl der Punkte ist, die nötig sind, um $\overline{\mathcal{C}}$ zu einer projektiven glatten Kurve zu komplettieren.*

Sei $\mathcal{F}$ eine Liftung des Frobenius-Endomorphismus $\phi_q$ auf $A^\dagger$. Dann induziert $\mathcal{F}$ einen Endomorphismus $\mathcal{F}^*$ auf den Kohomolgiegruppen. Der Hauptsatz der Monsky-Washnitzer Kohomologie ist der folgende Fixpunktsatz von Lefschetz.

**Satz 2.4.11 (Fixpunktsatz von Lefschetz)** *Sei $\overline{\mathcal{C}}/\mathbb{F}_q$ eine glatte affine Kurve über $\mathbb{F}_q$. Dann ist die Anzahl der $\mathbb{F}_{q^k}$-rationalen Punkte von $\overline{\mathcal{C}}$ gleich*

$$\#\overline{\mathcal{C}}(\mathbb{F}_{q^k}) = \mathrm{Tr}\left(q^k\mathcal{F}^{-k*}; H^0(\overline{\mathcal{C}}, \mathbb{Q}_q)\right) - \mathrm{Tr}\left(q^k\mathcal{F}^{-k*}; H^1(\overline{\mathcal{C}}, \mathbb{Q}_q)\right).$$

Da $H^0(\overline{\mathcal{C}}, \mathbb{Q}_q)$ ein eindimensionaler Vektorraum ist, auf dem $\mathcal{F}^*$ trivial operiert, schliessen wir, dass gilt

$$\mathrm{Tr}\left(q^k\mathcal{F}^{-k*}; H^0(\overline{\mathcal{C}}, \mathbb{Q}_q)\right) = q^k.$$

Um die $\mathbb{F}_{q^k}$-rationalen Punkte von $\overline{\mathcal{C}}$ zu ermitteln reicht es also, die Operation von $\mathcal{F}^*$ auf $H^1(\overline{\mathcal{C}}, \mathbb{Q}_q)$ zu berechnen.

Die auf Monsky-Washnitzer Kohomologie basierenden Algorithmen verfahren also in zwei Schritten: erst wird eine hinreichend genaue Approximation der Liftung $\mathcal{F}$ berechnet und dann wird eine Basis von $H^1(\overline{\mathcal{C}}, \mathbb{Q}_q)$ konstruiert, zusammen mit Reduktionsformeln für die Darstellung von Differentialformen auf dieser Basis. Für weitere algorithmische Details, siehe [CoF$^+$06], Abschnitt 17.3.

**Bemerkung 2.4.12** 1999 führte Satoh den ersten $p$-adischen Ansatz durch Serre-Tate Lifung für elliptische Kurven ein. 2001 gelang es Kedlaya, das Verfahren unter Verwendung der Monsky-Washnitzer Kohomologie auf hyperelliptische Kurven beliebigen Geschlechts zu verallgemeinern, indem er eine rigide analytische Liftung anstelle der kanonischen Liftung verwendete.

Ein besonders vorteilhafter Aspekt der Methode von Kedlaya ist, dass es offensichtlich keine theoretischen Hindernisse bei der Generalisierung auf größere Klassen von Kurven gibt. Diese Beobachtung führte bald zu Punktezählalgorithmen für superelliptische Kurven [GaG01], $C_{ab}$ Kurven [DeV06] und Kurven, die nichtdegeneriert bezüglich ihres Newton-Polytops sind [CDV06].

Die derzeit schnellsten Methoden für glatte elliptische Kurven benötigen $O(k^{2+\varepsilon})$ Zeit und $O(k^2)$ Speicher, um die $\mathbb{F}_{p^k}$-rationalen Punkte einer elliptischen Kurve zu zählen, siehe [Ler03] für eine Übersicht. Mit der Methode von Kedlaya kann man die $\mathbb{F}_{p^k}$-rationalen Punkte einer glatten hyperelliptischen Kurve in $O(g^{4+\varepsilon}k^{3+\varepsilon})$ Zeit und $O(g^3k^3)$ Speicher berechnen. In der Komplexitätsbetrachtung wird $p$ als Konstante gewertet, da diese Algorithmen nur für $p \leq 5$ akzeptable Laufzeiten erbringen. Harvey ist es 2006 gelungen, Kedlayas Algorithmen auf große Charakteristik zu übertragen, man kann nun die $\mathbb{F}_p$-rationalen Punkte einer glatten hyperelliptischen Kurve in $O(\sqrt{p}g^3)$ Schritten berechnen, siehe [Hav06].

## 2.5 Endomorphismus Methoden: der Cartier Operator

In Abschnitt 2.2 haben wir veranschaulicht, dass das charakteristischen Polynom $\chi_{\phi_p}$ des Frobenius-Endomorphismus eine Aussage über die Ordnung der Jacobischen Varietät der zugrundeliegenden Kurve ermöglicht und $p$-adische-

und $l$-adische Methoden zur Approximation dieses Polynoms vorgestellt. Anstatt das charakteristische Polynom zu approximieren, besteht natürlich auch die Möglichkeit, $\chi_{\phi_p}$ aus der Darstellungsmatrix eines geeigneten Morphismus direkt auszurechnen. Daher betrachten wir nun das Dual des Frobenius auf $H^0(X, \Omega_X)$, den Cartier-Operator $\mathcal{C}$. Dies ist eine Abbildung, die sich mit Hilfe der Hasse-Witt Matrix explizit berechnen läßt und die Angabe von $\chi_{\phi_p}$ modulo $p$ ermöglicht.

Wir führen nun grundlegende Eigenschaften des Cartier-Operators ein und zeigen, wie sich $\mathcal{C}$ auf den holomorphen Differentialen $\Omega^1(X)$ der Kurve berechnen läßt und wie man aus der Kenntnis von $\chi_{\phi_p}$ modulo einer Zahl $m$ das Polynom über $\mathbb{Z}$ erreichnen kann. Danach betrachten wir unterschiedliche Verfahren zur Ermittlung einer Basis von $\Omega^1(X)$. Ein generisches Verfahren zur Bestimmung der Hasse-Witt Matrix, welches ohne die Basisberechnung von $\Omega^1(X)$ auskommt, wird anschliessend vorgestellt. Zuletzt fassen wir einige Ergebnisse zusammen, die eine Berechnung der Ordnung der Jacobischen aus der Hasse-Witt Matrix ermöglichen und auf der Bestimmung einiger weniger Koeffizienten einer hohen Polynompotenz basiert.

### 2.5.1 Definition und grundlegende Eigenschaften

Sei $X$ ein Schema, dessen lokale Ringe alle Charakteristik $p > 0$ haben. Nach Definition 2.2.1 ist der absolute Frobenius-Endomorphismus $\phi_p$ die Abbildung, die trivial auf dem topologischen Raum operiert und $\phi_p^\sharp : \mathcal{O}_X \to \mathcal{O}_X$ die Abbildung, die jedes Element zur $p$-ten Potenz erhebt.

Ist $X$ eine projektive, glatte Kurve über einem vollkommenen Körper $K$ der Charakterisitk $p > 0$, so induziert $\phi_p$ eine Abbildung

$$\phi_p^* : H^1(X, \mathcal{O}_X) \to H^1(X, \mathcal{O}_X)$$

auf der Kohomologie. Diese Abbildung ist nicht linear, aber *p-linear*, es gilt also $\phi_p^*(\lambda a) = \lambda^p \phi_p^*(a)$ für alle $\lambda \in H^1(X, \mathcal{O}_X)$.

Als Konsequenz der Serre-Dualität erhält man für projektive, glatte Kurven, dass $H^1(X, \mathcal{O}_X)$ und $H^0(X, \Omega_X)$ zueinander duale Vektorräume sind (s. [Har77], III.7.12.2).

**Definition 2.5.1** *Die durch das Dual des Frobenius-Operators induzierte Abbildung $\mathcal{C} : H^0(X, \Omega_X) \to H^0(X, \Omega_X)$ nennt man Cartier-Operator (zu $\phi_p$) von $X$.*

Der Cartier-Operator operiert also auf den holomorphen Differentialen $\Omega^1(X)$. Sei d die kanonische Derivation des Funktionenkörpers $K(X)$. Dann lässt sich für ein separierendes Element $u$ jedes $z\mathrm{d}u \in \Omega^1(X)$ darstellen, als

$$z \ \mathrm{d}u \quad = \quad (z_0^p + z_1^p u + \ldots + z_{p-1}^p u^{p-1}) \ \mathrm{d}u \tag{2.8}$$

mit rationalen Funktionen $z_1, \ldots z_{p-1}$. Der Cartier-Operator hat damit auf $\Omega^1(X)$ die Form

$$\mathcal{C}(z \ \mathrm{d}u) = z_{p-1}\mathrm{d}u. \tag{2.9}$$

TATE hat gezeigt, dass diese Form unabhängig von der Wahl des separierendem Element $u$ ist (s. [Tat52]). Desweiteren hat CARTIER die folgende Liste von Eigenschaften des Cartier-Operators erstellt (s. [Ser58]):

$$\begin{aligned}
\mathcal{C}(z_1+z_2) &= \mathcal{C}(z_1)+\mathcal{C}(z_2) \quad \forall z_1,z_2\in\Omega^1(X) && (2.10)\\
\mathcal{C}(h^p z) &= h\mathcal{C}(z) \quad \text{für } h\in K(X), z\in\Omega^1(X) && (2.11)\\
\mathcal{C}(z) &= 0 \iff z=\mathrm{d}h \text{ mit } h\in K(X) \text{ und } z\in\Omega^1(X) && (2.12)\\
\mathcal{C}(z) &= z \iff z=\mathrm{d}h/h \text{ für } h\in K(X) \text{ und } z\in\Omega^1(X) && (2.13)\\
\mathcal{C}(h^{n-1}\mathrm{d}h) &= \begin{cases} \mathrm{d}h & \text{falls } n=p \\ 0 & \text{sonst} \end{cases} \quad \text{für } h\in K(X) && (2.14)
\end{aligned}$$

Sei $g$ das Geschlecht der Kurve $X$ und $\{z_1,\ldots,z_g\}$ eine Basis von $H^0(X,\Omega_X)$. Der Cartier-Operator operiert $K$-linear auf $H^0(X,\Omega_X)$ und daher kann die Abbildung durch die Matrix $A=(a_{ij})$ repräsentiert werden, die durch

$$\mathcal{C}(z_i)=\sum_{j=1}^{g} a_{ij}z_j \quad \text{mit } a_{ij}\in K \tag{2.15}$$

bestimmt ist.

**Definition 2.5.2** *Die Matrix $H=(a_{ij}^p)$ heißt **Hasse-Witt Matrix** der Basis $z_1,\ldots,z_g$.*

Ist $\{w_1,\ldots,w_g\}$ eine weitere Basis von $H^0(X,\Omega_X)$, die durch $w_i=\sum_{j=1}^g b_{ij}z_j$ mit $(b_{ij})\in GL_g(K)$ aus der obigen Basis überführt wird, dann ergibt sich die dazu korrespondierende Hasse-Witt Matrix $\hat{H}$ durch die Transformation

$$\hat{H}=(b_{ij})H(b_{ij}^p)^{-1}. \tag{2.16}$$

Dies spiegelt die $\frac{1}{p}$-Linearität des Cartier-Operators wieder, wie man sie aus der obigen Liste der Eigenschaften des Cartier-Operators entnehmen kann. Sei nun $\tilde{X}: f(u,v)=0$ eine affiner Schnitt von $X$ und $z\mathrm{d}u$ wie in Gleichung (2.8), dann gilt $\frac{\mathrm{d}^{p-1}z}{\mathrm{d}u^{p-1}}=-z_{p-1}^p$ und nach (2.9)

$$\mathcal{C}(z\ \mathrm{d}u) = \left(-\frac{\mathrm{d}^{p-1}z}{\mathrm{d}u^{p-1}}\right)^{\frac{1}{p}}\ \mathrm{d}u \tag{2.17}$$

Hat $z\mathrm{d}u$ die Form $z\mathrm{d}u=h\mathrm{d}u/g$ mit $g,h\in K(\tilde{X})$, dann gilt wegen der $\frac{1}{p}$-Linearität von $\mathcal{C}$

$$\mathcal{C}\left(\frac{h\mathrm{d}u}{g}\right) = \left(-\frac{\mathrm{d}^{p-1}(g^{p-1}h)}{\mathrm{d}u^{p-1}}\right)^{\frac{1}{p}}\frac{\mathrm{d}u}{g}. \tag{2.18}$$

MANIN hat in [Man65] gezeigt, wie die Hasse-Witt Matrix mit der Operation des Frobenius-Endomorphismus auf den $p$-Torsionsanteil der Jacobischen zusammenhängt. Wir fassen die Hauptaussage in dem folgenden Satz zusammen.

**Satz 2.5.3 (Manin)** *Sei $C$ eine Kurve von Geschlecht $g > 0$ über $\mathbb{F}_{p^n}$, die ein nichtspezielles System von $g$ Punkten über $\mathbb{F}_{p^n}$ enthält. Sei $H$ die Hasse-Witt Matrix von $C$ und $H_\pi = HH^{(p)} \cdots H^{(p^{n-1})}$, sei $\mathcal{K}(t)$ das charakteristische Polynom von $H_\pi$ und $\chi_{\phi_p}(t)$ das charakteristische Polynom des Frobenius-Endomorphismus $\phi_p$ der Jacobischen von $C$. Dann gilt*

$$\chi_{\phi_p}(T) \equiv (-1)^g t^g \mathcal{K}(T) \ mod\ p.$$

Die Notation $H^{(s)}$ bedeutet hier: Erhebung der Matrixeinträge zur $s$-ten Potenz. Wir erhalten also eine Identifikation des charakteristischen Polynoms des Frobenius-Endomorphismus modulo $p$. Aus Abschnitt 2.2 wissen wir, dass gilt

$$\#J_C(\mathbb{F}_p) = \chi_{\phi_p}(1).$$

Aus der Kenntnis der Hasse-Witt-Matrix $H$ können wir also die Gruppenordnung von $J_C(\mathbb{F}_p)$ modulo $p$ ableiten. Das *Hasse-Weil-Intervall* grenzt die Gruppenordnung weiter ein

$$\lceil(\sqrt{p}-1)^{2g}\rceil \leq \#J_C(\mathbb{F}_p) \leq \lfloor(\sqrt{p}+1)^{2g}\rfloor \tag{2.19}$$

Es gilt

$$\left(\chi_{\phi_p}(\phi_p)\right)(D) = 0 \text{ für alle } D \in J_C(\mathbb{F}_p).$$

Hat $J_C(\mathbb{F}_p)$ eine effiziente Arithmetik, so kann man aus der Kenntnis von $\chi_{\phi_p}(T)$ modulo $p$ mit einem Schoof-ähnlichen- und/oder einem Baby-Step/Giant-Step Algorithmus $\chi_{\phi_p}$ über $\mathbb{Z}$ bestimmen, wie dies z.B. in [GaH00] und [CMT02] für hyperelliptische Kurven von Geschlecht 2 durchgeführt wurde.

**Beispiel 2.5.4 (Der MCT-Algorithmus)** Im Jahre 2002 zeigten Matsuo, Chao und Tsujii, wie man aus der Kenntnis des charakteristischen Polynoms $\chi_{\phi_q}(T)$ modulo $m$ für $m \in \mathbb{N}$ im Falle hyperelliptischer Kurven von Geschlecht 2 mit einem Baby-Step/Giant-Step Algorithmus das Polynom $\chi_{\phi_q}(T)$ über $\mathbb{Z}$ rekonstruieren kann. Die Grundzüge dieses Algorithmus wollen wir in diesem Beispiel darstellen.

Sei $C$ eine hyperelliptische Kurve von Geschlecht 2 über einem endlichen Körper $\mathbb{F}_q$ der Charakteristik $p$ mit $q = p^n$ Elementen und $n > 2$. Das charakteristische Polynom $\chi_{\phi_q}(T)$ des Frobenius-Endomorphismus hat die Form

$$\chi_{\phi_q}(T) = T^4 - s_1T^3 + s_2T^2 - qs_1T + q^2,$$

mit ganzen Zahlen $s_1$ und $s_2$. Die Gruppenordnung von $J_C(F_q)$ ist also gegeben durch

$$\#J_C(\mathbb{F}_q) = \chi_{\phi_q}(1) = q^2 + 1 - s_1(q+1) + s_2.$$

Nach dem Satz von Weil haben die Nullstellen von $\chi_{\phi_q}(T)$ Absolutbetrag $\sqrt{q}$ und daraus folgen unmittelbar Schranken für $s_1$ und $s_2$. Nutzt man zudem, dass diese Nullstellen als Paare komplex konjugierter Zahlen auftreten, erhält man eine noch schärfere Abschätzung

$$|s_1| \leq 4\sqrt{q} \quad \text{und} \quad 2|s_1|\sqrt{q} - 2q \leq s_2 \leq \frac{s_1^2}{4} + 2q. \tag{2.20}$$

Wir gehen nun davon aus, dass das charakteristische Polynom $\chi_{\phi_q}(T)$ des Frobenius-Endomorphismus modulo einer ganzen Zahl $m$ bekannt ist, das heißt: $s_1$ und $s_2$ sind modulo $m$ bekannt. Wir führen neue Variablen für den bekannten und unbekannten Teil von $s_1$ und $s_2$ ein

$$s_1 = \overline{s_1} + m\tilde{s_1} \quad \text{und} \quad s_2 = \overline{s_2} + m\tilde{s_2}.$$

Unser Ziel ist es also, $\tilde{s_1}$ und $\tilde{s_2}$ zu finden. Daher wählen wir einen zufälligen Divisor $D \in Pic_C^0(\mathbb{F}_q) \simeq J_C(\mathbb{F}_q)$ und hoffen, dass die Ordnung des Divisors groß genug für unsere Zwecke ist. Der Fall, dass $J_C(\mathbb{F}_q)$ hochgradig nicht-zyklisch ist, tritt in der Praxis selten auf und ist leicht behandelbar.

Die Ordnung von $D$ teilt die Gruppenordnung $\chi_{\phi_q}(1)$, daher haben wir die Gleichung

$$(q^2 + 1 - \overline{s_1}(q+1) + \overline{s_2}) \cdot D + (-\tilde{s_1}(q+1) + \tilde{s_2}) \cdot m \cdot D = 0.$$

Um einen Baby-Step/Giant-Step Algorithmus zu erhalten, trennt man üblicherweise die beiden Unbekannten – auf jede Seite der Gleichung eine. Hier liegen die Unbekannten $\tilde{s_1}$, $\tilde{s_2}$ in Intervallen verschiedener Größe und daher ist es notwendig, $\tilde{s_2}$ nochmal in zwei Unbekannte aufzuteilen. Sei $\nu \in \mathbb{Z}$ ein Parameter, den wir nachher festlegen werden. Wir schreiben $\tilde{s_2} = t_2 + \nu u_2$ mit $0 \leq t_2 < \nu$. Dann können wir $\chi_{\phi_q}(1) \cdot D = 0$ darstellen, als

$$\left(q^2 + 1 - \overline{s_1}(q+1) + \overline{s_2} + m(-\tilde{s_1}(q+1) + \nu u_2)\right) \cdot D = -t_2 m \cdot D. \qquad (2.21)$$

Der Algorithmus geht nun wie folgt vor: zuerst wird die rechte Seite der Gleichung (2.21) berechnet und in einer Datenstruktur abgespeichert, die leicht durchsucht werden kann. Dann wird die linke Seite von (2.21) für alle möglichen Werte von $\tilde{s_1}$ und $u_2$ berechnet, bis eine Übereinstimmung mit der rechten Seite erzielt wurde. Für jedes $s_1$ wird die Abschätzung aus (2.20) benutzt, um den Bereich der möglichen Werte für $u_2$ zu bestimmen. Eine genaue Untersuchung des Suchbereichs aus (2.20) führt zu einem optimalen Wert von $\nu \approx q^{3/4}/m$, was zu einer Laufzeit von etwa $O(q^{3/4}/m)$ Operationen in $Pic_C^0(\mathbb{F}_q)$ führt. Dieser Algorithmus wurde von Gaudry und Schost in einen Algorithmus ohne große Speicheranforderungen überführt, siehe [GaS04].

**Bemerkung 2.5.5** Nach ([BGS07], S. 1800-1801) hat der MCT-Algorithmus im generellen Fall eine Komplexität von $O(q^{(\frac{g}{2}-\frac{1}{4})}/\sqrt{m})$ für eine Kurve von Geschlecht $g$, abgesehen von $g = 2$ . Wir interessieren uns besonders für den Fall $g = 4$, $q = m = p$, für den wir eine Komplexität von $O(p^{\frac{5}{4}})$ erhalten.

### 2.5.2 Basisberechnung für Riemann-Roch Räume

In Abschnitt 2.5.1 haben wir gesehen, dass der Cartieroperator auf den holomorphen Differentialen $\Omega^1(\tilde{X})$ operiert. Können wir eine $K$-Basis für $\Omega^1(X)$ berechnen, dann lässt sich der Cartieroperator mit der Hasse-Witt Matrix beschreiben. Eine Möglichkeit, diese Basis zu berechnen, ist über die *Riemann-Roch Räume*

$$\mathcal{L}(D) := \{a \in K(\tilde{X}) : (a) \geq -D\} \cup \{0\}$$

für einen Divisor $D$ von $K(\tilde{X})$. Seien die Bezeichnungen so wie in (2.17), dann ist $(\mathrm{d}u)$ ein kanonischer Divisor von $K(\tilde{X})$ und wir können jedes $\omega \in \Omega(\tilde{X})$ darstellen, als $\omega = b\mathrm{d}x$ mit $b \in K(\tilde{X})$. Es existiert dann ein $K$-Vektorraum Isomorphismus

$$\begin{array}{rcl} \mathcal{L}\left((\mathrm{d}u) - D\right) & \xrightarrow{\sim} & \Omega_{\tilde{X}}(D) \\ b & \mapsto & b\,\mathrm{d}u \end{array}$$

mit

$$\Omega_{\tilde{X}}(D) := \{\omega \in \Omega(\tilde{X}) : (\omega) \geq -D\},$$

$\Omega(\tilde{X})$ ist dabei der Raum der meromorphen Differentialformen (s.[Sti93], I.5.14. und [Hes02] Abschnitt 9). Für $D = 0$ erhalten wir

$$\mathcal{L}\left((\mathrm{d}u)\right) \cong \Omega_{\tilde{X}}(0) = \Omega^1(\tilde{X}).$$

Somit ist es möglich, eine Basis der holomorphen Differentiale mit Hilfe von Riemann-Roch Räumen zu berechnen.

Es gibt zahlreiche Methoden zur Basisberechnung von Riemann-Roch Räumen. F. Hess gibt in [Hes02] einen umfangreichen Überblick und Referenzen zu den unterschiedlichen Ansätzen. Er unterscheidet zwischen *geometrischen* und *arithmetischen Methoden*. Die geometrischen Methoden benutzen adjungierte Formen und den Satz von Brill-Noether. Die arithmetischen Methoden basieren auf Idealen von Bewertungsringen und behandeln üblicherweise aufwendige Potenzreihenentwicklungen algebraischer Funktionen an speziellen Stellen. F. Hess hat in [Hes02] eine effiziente Methode vorgestellt, die direkt auf den Bewertungsringen und deren Idealen rechnet.

Weitere Details, sowie Beispiele und Anwendungen findet man in [Hes02]. Der Algorithmus ist in KANT implementiert, siehe [DFK+97]. Die Komplexität des Algorithmus setzt sich wie folgt zusammen: Sei $d := [K(\tilde{X}) : K(u)]$. Für einen Divisor $D$ von $K(\tilde{X})$ sei die *Höhe* von $D$ definiert, als die Summe der Grade des Pol- und Nulldivisors in $D$, also $h(D) := \deg_K(D)_0 + \deg_K(D)_\infty$. Wir betrachten den affinen Schnitt $f(u,v) = 0$ von $X$ als Polynom $f(v) := v^d + a_1 v^{d-1} + \cdots + a_d \in (K[u])[v]$ über $K[u]$ und definieren damit

$$C_f := \max\{\lceil \deg_u(a_i)/i \rceil : 1 \leq i \leq d\}.$$

Der Algorithmus in [Hes02] berechnet dann eine $K$-Basis von $\mathcal{L}(D)$ in

$$O\left(\log(h(D)) \cdot |\mathrm{supp}(D)| \cdot (dC_f\eta)^\alpha)\right)$$

Operationen, wobei $\eta$ der maximale Grad aller Stellen in $D$ ist, sowie $\alpha \in \mathbb{R}^{>0}$.

### Geometrische Methoden

Sei $\nabla$ der Operator $\frac{\partial^{2p-2}}{\partial u^{p-1}\partial v^{p-1}} : K[u,v] \longrightarrow K[u^p, v^p]$, wobei $\frac{\partial^{a+b}}{\partial u^a \partial v^b}$ die partiellen Hasse-Schmidt Ableitungen bezeichnet. Dann haben wir die wichtige, von K.-O. Stöhr und J. F. Voloch in [StV87] aufgestellte Gleichung

$$\mathcal{C}\left(\frac{h\mathrm{d}u}{f_v}\right) \;=\; (\nabla(f^{p-1}h))^{\frac{1}{p}}\frac{\mathrm{d}u}{f_v}. \tag{2.22}$$

Diese Gleichung erlaubt eine direkte Darstellung der Differentialbasis durch adjungierte Formen zur Berechnung der Hasse-Witt Matrix. Die Operation von $\nabla$ kann beschrieben werden als

$$\nabla\left(\sum_{i,j} c_{ij}u^i v^j\right) \quad = \quad \sum_{i,j} c_{ip+p-1,jp+p-1}u^{ip}v^{ip}. \tag{2.23}$$

Sei $z_1 \ldots, z_g$ eine Basis der holomorphen Differentiale. Als eine Konsequenz des Brill-Noether Residuensatzes (s. [Bri88]) können wir adjungierte Formen $h_1, \ldots, h_g$ der Ordnung $d-3$ finden, so dass gilt

$$z_i \quad = \quad h_i(u,v)f_v^{-1}\mathrm{d}u \quad i = 1, \ldots, g \tag{2.24}$$

Diese Polynome $h_1, \ldots, h_g$ nennen wir im folgenden eine *Basis der adjungierten Formen*. Nach Gleichung (2.22) bekommen wir

$$\nabla(f^{p-1}h_i) \quad = \quad \sum_{1}^{g} a_{ij}^p h_j^p, \tag{2.25}$$

wobei $A = (a_{ij}^p)$ die Hasse-Witt Matrix ist. Damit haben wir also einen sehr einfachen Weg gefunden, die Hasse-Witt Matrix aus den Daten $f^{p-1}, h_1, \ldots, h_g$ zu berechnen.

**Beispiel 2.5.6** Sind die adjungierten Formen Monome, dann sind die Einträge der Hasse-Witt Matrix bestimmte Koeffizienten von $f^{p-1}$. Betrachten wir zum Beispiel eine elliptische oder hyperelliptische Kurve

$$v^2 = q(u),$$

wobei $q(u) \in k[u]$, char $k \geq 3$ ein Polynom von Grad $2g+1$ ohne doppelte Nullstellen ist. Dann ist eine Basis der adjungierten Formen von Ordnung $d-3$ gegeben, durch $u^{i-1}$, $i = 1, \ldots, g$. Wir benutzen Gleichung (2.23) und erhalten

$$\frac{\partial^{2p-2}}{\partial u^{p-1}\partial v^{p-1}}((v^2-q(u))^{p-1}u^{i-1}) = -\frac{\partial^{p-1}}{\partial v^{p-1}}(q(u)^{\frac{p-1}{2}}u^{i-1}) = \sum_j a_{pj-i}u^{pj-p}$$

mit $q(u)^{\frac{p-1}{2}} = \sum_{j=0}^{\infty} a_j u^j$. Wir erhalten als Hasse-Witt Matrix

$$H = \begin{pmatrix} a_{p-1} & a_{2p-1} & \cdots & a_{gp-1} \\ a_{p-2} & a_{2p-2} & \cdots & a_{gp-2} \\ \vdots & \vdots & & \vdots \\ a_{p-g} & a_{2p-g} & \cdots & a_{gp-g} \end{pmatrix}.$$

Dies ist die klassische Formel für die Hasse-Witt Matrix (s. [Yui78], S.381). Das selbe lässt sich für alle Kurven mit reiner Gleichung $y^n = q(x)$, $n \in \mathbb{N}$ wiederholen. Siehe z.B. [BTW05] für Picardkurven.

### 2.5.3 Der Cartier-Operator auf $H^2(\mathbb{P}^2, \mathcal{O}(-deg\ X))$

In diesem Abschnitt stellen wir eine generische Methode zur Bestimmung der Hasse-Witt Matrix vor, bei der wir keine Basis vorberechnen müssen. Diese wurde ursprünglich von Bouw in [Bou98] für Quartiken verwendet und soll hier auf eine absolut irreduzible, reduzierte, projektive ebene Kurve $X$ verallgemeinert werden. Sei $X$ durch das homogene Polynom $F$ von Grad $d$ gegeben. Die „negativen“ Monome

$$B_d := \{x^{l_0} y^{l_1} z^{l_2} \ : \ \forall_{i=0..2}\ l_i < 0,\ l_0 + l_1 + l_2 = -d\}$$

bilden eine Basis von $H^2(\mathbb{P}^2, \mathcal{O}(-d))$ (s. [Har77], III.5.1.(c)).

**Satz 2.5.7** *Sei $B_{deg\ X} = \{\gamma_1, \ldots, \gamma_g\}$. Der $(i,j)$-te Eintrag der Hasse-Witt Matrix ist der Koeffizient von $\gamma_j$ in $F^{(p-1)}\gamma_i^p$.*

**Beweis:** Die Idealgarbe $\mathcal{I}_X$ von $X$ ist isomorph zu $\mathcal{O}_{\mathbb{P}}(-deg\ X)$ (S. [Har77] II.6.17, II.6.18). Wir haben eine kurze exakte Sequenz

$$\begin{array}{ccccccccc} 0 & \longrightarrow & \mathcal{I}_X & \xrightarrow{\cdot F} & \mathcal{O}_{\mathbb{P}} & \longrightarrow & \mathcal{O}_{\mathbb{P}}/\mathcal{I}_X & \longrightarrow & 0 \\ & & \downarrow \wr & & & & \downarrow \wr & & \\ & & \mathcal{O}_{\mathbb{P}}(-deg\ X) & & & & \mathcal{O}_X & & \end{array}$$

Daraus erhalten wir eine lange exakte Sequenz

$$\cdots \to H^1(\mathbb{P}^2, \mathcal{O}_{\mathbb{P}}) \to H^1(X, \mathcal{O}_X) \to H^2(\mathbb{P}^2, \mathcal{O}_{\mathbb{P}}(-deg\ X)) \to H^2(\mathbb{P}^2, \mathcal{O}_{\mathbb{P}}) \to \cdots$$

So ergibt sich der Isomorphismus

$$H^1(X, \mathcal{O}_X) \to H^2(\mathbb{P}^2, \mathcal{O}_{\mathbb{P}}(-deg\ X)),$$

wenn wir zeigen können $H^i(\mathcal{O}_{\mathbb{P}}) = 0$ für $i = 1, 2$. Für $i = 1$ gilt das nach ([Har77], II.5.1.(6)). $H^i(\mathcal{O}_{\mathbb{P}}) = 0$ folgt aus Serre-Dualität:

$$H^2(\mathcal{O}_{\mathbb{P}}) = H^2(\Omega^0_{\mathbb{P}}) \cong H^0(\Omega^2_{\mathbb{P}}) \cong H^0(\mathcal{O}_{\mathbb{P}}(-3)).$$

Die letzte Isomorphie ergibt sich aus ([Har77], II.8.20.1), da $\Omega^2_{\mathbb{P}}$ die kanonische Garbe von $\mathbb{P}^2$ ist. Es gilt aber $H^0(\mathcal{O}_{\mathbb{P}}(-3)) \cong \Gamma(\mathbb{P}^2, \mathcal{O}_{\mathbb{P}}(-3))$ nach ([Har77], II.1.4.) und diese Menge besteht nur aus Null, wegen $\Gamma_*(\mathbb{P}, \mathcal{O}_{\mathbb{P}}(-3)) = k[x, y, z]$ ([Har77] II.5.13).

Nun können wir mit dieser Einbettung die Operation des Frobenius berechnen. Ist $\phi_p$ der Frobeniusmorphismus auf $\mathbb{P}^2$, dann bildet $\phi_p^*\ \mathcal{O}_X$ auf $\mathcal{O}_{X^p}$ ab, wobei $X^p$ das Unterschema von $\mathbb{P}^2$ ist, das durch $F^p = 0$ definiert wird. Da $X$ ein abgeschlossenes Unterschema von $X^p$ ist, erhalten wir ein kommutatives Diagramm

$$\begin{array}{ccccccccc} 0 & \longrightarrow & \mathcal{O}_{\mathbb{P}}(-(deg\ X)p) & \longrightarrow & \mathcal{O}_{\mathbb{P}} & \longrightarrow & \mathcal{O}_{X^P} & \longrightarrow & 0 \\ & & \downarrow{\scriptstyle \cdot F^{(p-1)}} & & \downarrow & & \downarrow & & \\ 0 & \longrightarrow & \mathcal{O}_{\mathbb{P}}(-deg\ X) & \longrightarrow & \mathcal{O}_{\mathbb{P}} & \longrightarrow & \mathcal{O}_X & \longrightarrow & 0. \end{array}$$

Dabei bedeutet „$\cdot F^{(p-1)}$“, dass sich die Abbildung durch Multiplikation eines Elementes aus $\mathcal{O}_{\mathbb{P}}(-(deg\ X)p)$ mit $F^{(p-1)}$ ergibt. Da $\mathcal{O}_{\mathbb{P}}(-(deg\ X)p)$ lokal durch $F^{-p}$ gegeben ist, ist die Abbildung wohldefiniert. Aus dem Diagramm erhalten wir über $H^1(X, \mathcal{O}_X) \stackrel{\phi^*}{\to} H^1(X, \mathcal{O}_X)$ das kommutative Diagramm

$$\begin{array}{ccc} H^1(X, \mathcal{O}_X) & \stackrel{\sim}{\longrightarrow} & H^2(\mathbb{P}, \mathcal{O}_{\mathbb{P}}(-deg\ X)) \\ \downarrow{\scriptstyle \phi_1^*} & & \downarrow{\scriptstyle \phi_1^*} \\ H^1(X, \mathcal{O}_{X^p}) & \stackrel{\sim}{\longrightarrow} & H^2(\mathbb{P}, \mathcal{O}_{\mathbb{P}}(-(deg\ X)p)) \\ \downarrow & & \downarrow{\scriptstyle \cdot F^{p-1}} \\ H^1(X, \mathcal{O}_X) & \stackrel{\sim}{\longrightarrow} & H^2(\mathbb{P}, \mathcal{O}_{\mathbb{P}}(-deg\ X)). \end{array}$$

Für ein Basiselement $\gamma_i \in B_{deg\ X}$ von $H^2(\mathbb{P}, \mathcal{O}_{\mathbb{P}}(-deg\ X))$ gilt nun $\phi_1^*(\gamma_i) = \gamma_i^p$ und dessen Bild in $H^2(\mathbb{P}, \mathcal{O}_{\mathbb{P}}(-deg\ X))$ ist $F^{p-1}\gamma_i^p$. □

Damit können wir für beliebige irreduzible Kurven die Hasse-Witt Matrix angeben.

**Beispiel 2.5.8** Für Geschlecht $g = 3$ sind die nichthyperelliptischen Kurven genau die Quartiken in $\mathbb{P}^2$ (s. [Har77], IV.5.2.1). Eine Basis von $H^2(\mathbb{P}, \mathcal{O}(-4))$ ist gegeben durch die negativen Monome

$$\gamma_1 = (x^2yz)^{-1}, \quad \gamma_2 = (xy^2z)^{-1}, \quad \gamma_3 = (xyz^2)^{-1}.$$

Sei $Q$ eine projektive Quartik und $q(X, Y)$ ihr affiner Schnitt bei $z = 1$. Ist $q(X, Y)^{(p-1)} = \sum_{i,j=0} a_{ij} X^i Y^j$, dann ist die Hasse-Witt Matrix gegeben, durch

$$H = \begin{pmatrix} a_{2p-2,p-1} & a_{p-2,2p-1} & a_{p-2,p-1} \\ a_{2p-1,p-2} & a_{p-1,2p-2} & a_{p-1,p-2} \\ a_{2p-1,p-1} & a_{p-1,2p-1} & a_{p-1,p-1} \end{pmatrix}.$$

Bouw hat in ihrer Dissertation für alle Quartiken mit nichttrivialer Automorphismengruppe die Einträge der Hasse-Witt Matrix als Binomialkoeffizienten angegeben, die bei der Potenzierung der definierenden Polynomgleichung auftreten (s. [Bou98], 4.4.4.).

Damit haben wir die Berechnung der Hasse-Witt Matrix von $H^1(X, \mathcal{O})$ auf $H^2(\mathbb{P}^2, \mathcal{O}(-deg\ X))$ verlagert, in der wir immer eine generische Basis kennen. Mit Hilfe von Cech-Kohomologie könnte man auch direkt in $H^1(X, \mathcal{O})$ eine Basis errechnen und dann die Hasse-Witt Matrix bestimmen.

### 2.5.4 Koeffizienten von Polynompotenzen

Wie wir im vorhergehenden Abschnitt festgestellt haben, läuft die Erstellung der Hasse-Witt Matrix in vielen Fällen auf die Berechnung einiger weniger Koeffizienten einer Polynompotenz hinaus. Dieses Kapitel gibt einen Überblick über mögliche Wege zur Berechnung dieser Koeffizienten mit Schwerpunkt auf kryptografische Anwendbarkeit.

Sei $F \in k[x,y]$ ein Polynom über einem Körper. Wir wollen den Koeffizienten $h_{ij}$ von $F(x,y)^l = \sum_{ij} h_{ij} x^i y^j$, $l \in \mathbb{N}$ bestimmen. Das explizite Ausrechnen des gesamten Polynoms gestaltet sich extrem aufwändig. Alagar und Probst haben in [Ala87] einen hybriden Algorithmus aus Karatsuba-Multiplikation und diskreter Fouriertransformation vorgestellt, die asymptotisch nur den Speicherplatz benötigt, um das Ergebnis zu speichern. Will man mit Karatsuba zwei Polynome multiplizieren, so benötigt man $O(s^{\lg 3})$ Schritte, wenn $s$ der höhere Grad der beiden Polynome ist.

Wendet man die Multinomialformel für spezielle Polynome an und summiert die gesuchte Monompotenz auf, wie Bouw in ([Bou98], 4.4.), so ist der Zeitaufwand mindestens quadratisch in $l$.

#### Der univariate Fall

Flajolet und Salvy berechnen in [FlS97] einen Koeffizienten eines univariaten Polynoms mit Hilfe einer Differentialgleichung. Sei $G(x) = \sum_i g_i x^i \in k[x]$ ein univariates Polynom und $H(x) = \sum_i h_i x^i := G(x)^l$. Ableiten von $H$ ergibt

$$H' = lG^{l-1}G'$$

und wir erhalten die Differentialgleichung

$$GH' = lHG'.$$

Wir leiten diese Gleichung $n$ mal ab

$$\sum_{i=0}^{n} \binom{n}{i} G^{(i)} H^{(n-i+1)} = \sum_{i=0}^{n} \binom{n}{i} G^{(i+1)} H^{(n-i)},$$

werten dann in null aus und erhalten mit dem Startwert $H^{(0)}(0) = G(0)^l$ eine lineare Rekurrenz für die Koeffizienten von $H$

$$G(0)H^{(n+1)}(0) = \sum_{i=0}^{n} \binom{n}{i} G^{(i+1)}(0) H^{(n-i)}(0) - \sum_{i=1}^{n} \binom{n}{i} G^{(i)}(0) H^{(n-i+1)}(0),$$

da $H^{(n+1)}(0)$ der $(n+1)$-te Koeffizient der Taylorentwicklung von $H$ ist. Da $G^{(s)}(0) = 0$ sobald $s > deg_x G$, erhält man einen Algorithmus mit linearem Zeitaufwand und vernachlässigbarem Speicher zur Berechnung eines Koeffizienten von $H$. Es gibt bereits einige Verbesserungen des Algorithmus von Flajolet/Salvy: in [Chu88] stellen Chudnovsky und Chudnovsky einen Baby-Step/Giant-Step Algorithmus vor, um das linksassoziierte Matrixprodukt

$$C = \prod_{i=1}^{\sqrt{l}} A(x+i), \quad A \in Mat_s(k[x])$$

in

$$O(s^\omega M(\sqrt{l}) + s^2 M(\sqrt{l})\log(l))$$

Operationen in $k$ zu berechnen. Dabei ist

$$M(d) = d\log(d)\log(\log(d))$$

die Zeitkomplexität, zwei Polynome vom Grad $d$ über den Schönhage-Strassen Algorithmus zu multiplizieren und $O(s^\omega)$, $\omega = \lg 7$ die Zeit mit dem Strassen-Algorithmus zwei $s \times s$ Matrizen zu multiplizieren. Man benötigt für den Algorithmus von Chudnovsky $O(s^2\sqrt{l} + \sqrt{l}\log(l))$ Speicher.

In [BGS04] wird dieser Algorithmus noch einmal verbessert. Bostan, Gaudry und Schost benutzen dabei Lagrange-Interpolation, um einen schnellen Shift auf Polynomauswertungen zu berechnen, d.h. ist ein Polynom $P(x) \in k[x]$ vom Grad $d$ und $r_0, \ldots, r_d, a \in k$ vorgegeben, so berechnen sie aus $P(r_0), \ldots, P(r_d)$ in $O(M(d)\log(d))$ Operationen den Shift $P(r_0+a), \cdots, P(r_d+a)$. Damit verschnellern sie den Algorithmus von Chudnovsky und Chudnovsky und erhalten einen Algorithmus mit dem man einzelne Glieder einer linearen Rekurrenz berechnen kann.

**Satz 2.5.9** *Sei $R$ ein Ring, $A$ eine $s \times s$ Matrix mit linearen Einträgen aus $R[x]$ und $U_0$ aus $R^s$. Angenommen $(U_i)$ ist eine Folge von Elementen aus $R^s$, die definiert ist durch die lineare Rekurrenz*

$$U_{i+1} = A(i+1)U_i, \text{ für alle } i \geq 0.$$

*Sei $l \in \mathbb{N}$ und $1, \ldots, 2\lceil\sqrt{l}\rceil + 1$ Einheiten in $R$. Dann kann der Vektor $U_l$ mit $O(s^\omega\sqrt{l} + s^2 M(\sqrt{l})\log(l))$ Operationen in $R$ berechnet werden, unter der Verwendung von $O(s^2\sqrt{l})$ Speicher.*

Der Satz wurde dann in [BGS04] auf die lineare Rekurrenz von Flajolet und Salvy angewendet. Sei $h_i$ der $i$-te Koeffizient von $H$ und $g_i$ der von $G$. Dann lässt sich diese lineare Rekurrenz in die folgende Darstellung überführen: Sei $U_i = (h_{i-d}, \ldots, h_i)^t$ und $A(i)$ die $d \times d$ Matrix

$$A(i) = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ \frac{g_d(dl-(i-d))}{g_0 i} & \ldots & \ldots & \ldots & \frac{g_1(l-i+1)}{g_0 i} \end{pmatrix}.$$

Der Startvektor $U_0 = (0, \ldots, 0, g_0^l)$ kann mit binären Multiplizieren in $O(\log(l))$ Schritten berechnet werden. Die lineare Rekurrenz entspricht dann $U_{(i+1)} = A(i+1)U_i$ und der gesuchte Koeffizient tritt als Komponente eines Vektors $U_{i'}$ auf. Da die Matrix $A$ rationale Funktionen als Einträge enthält und man bei char $k > 0$ Gefahr läuft, durch null zu teilen, muss die Rekurrenz noch etwas angepasst werden (s. [BGS04]).

Aus Beispiel 2.5.6 wissen wir, dass sich die Hasse-Witt Matrix einer hyperelliptischen Kurve der Form $y^2 = g(x)$ aus speziellen Koeffizienten von $g^{(p-1)/2}$ ablesen lässt. Daher kann man mit der obigen Methode die Hasse-Witt Matrix einer hyperelliptischen Kurve berechnen (s.[BGS04]):

**Satz 2.5.10** *Sei $p$ eine Primzahl $n \geq 0$ und $\mathcal{C}$ eine hyperelliptische Kurve von Geschlecht $\gamma$ über $\mathbb{F}_{p^n}$, die durch die Gleichung $y^2 = g(x)$ definiert ist und $g(x)$ habe Grad $2\gamma + 1$. Ist $\gamma < p$, dann kann man die Hasse-Witt Matrix von $\mathcal{C}$ mit einer Komplexität von*

$$O\left(\gamma^{\omega+1}\sqrt{p} + \gamma^3 M(\sqrt{p})\log(p)M(n\gamma\log(p))\right)$$

*Bitoperationen und $O\left(n\gamma^3\sqrt{p}\log(p)\right)$ berechnen.*

Dieses Verfahren kann man sofort auf alle superelliptischen Kurven $y^n = f(x)$ verallgemeinern, wie es z.B. Bauer, Teske und Weng in [BTW05] für Picardkurven getan haben. Man beachte, dass diese Verfahren vom Grad der Kurve abhängen, nicht vom Geschlecht. Korollar III.10.4 aus [Sti93] erlaubt uns, die Komplexität im Geschlecht durch die Komplexität im Grad der Kurve abzuschätzen.

**Satz 2.5.11** *Sei $F = k(u, v)$ ein algebraischer Funktionenkörper. Dann haben wir die folgende Abschätzung für das Geschlecht $g$ von $F|k$:*

$$g \leq ([F : k(u)] - 1) \cdot ([F : k(v)] - 1).$$

Damit formulieren wir in den Bezeichungen von Abschnitt 2.5.1

**Korollar 2.5.12** $O(g) \subset O(deg_u f \cdot deg_v f)$.

### Der bivariate Fall

Will man dieses Konzept auf ein Polynom $F(x, y)$ in zwei Unbestimmten verallgemeinern, so erhält man weit aufwändigere Algorithmen. Die Idee von Flajolet und Salvy kann man mit

$$H(x, y) := F(x, y)^l = \sum_{ij} h_{ij} x^i y^j$$

in zwei Variablen durchführen, erhält dann aber eine quadratische Zeitkomplexität in $l$ und zudem noch einen linearen Speicheraufwand in $l$. Dieser entsteht dadurch, dass man nun nicht mehr ein Folgeglied nach dem anderen berechnen kann, sondern für einen Koeffizienten $h_{st}$ von $H$ alle $h_{ij}$ mit $0 \leq i \leq t$ $0 \leq j \leq deg_y F$ abspeichern muss. Wir stellen nun eine Möglichkeit vor, mit der man das bivariate Problem als Univariates betrachten kann.

### Lagrange-Interpolation

Man kann $F$ als Polynom $F(y) \in k[x][y]$ auffassen und Satz 2.5.9 mit $R = k[x]$ anwenden. Da die Polynomgrade in $x$ aber immer weiter ansteigen, kostet eine Operation in $R$ im schlechtesten auftretenden Fall $O(M(l))$ Operationen in $k$, daher betrachten wir nun einen Weg, der die Polynome in $R$ vor Anwendung der Arithmetik in einem Punkt auswertet.

In der obigen Terminologie ist $\mathcal{H} := H^{(t)}(0, y)$ hier ein Polynom in $y$. Betrachten wir das Lagrange-Interpolationspolynom für $\mathcal{H}$

$$\mathcal{H} = \sum_{i=0}^{deg\ \mathcal{H}} \mathcal{H}(i) \frac{\prod_{j=0, j\neq i}^{deg\ \mathcal{H}} (y-j)}{\prod_{j=0, j\neq i}^{deg\ \mathcal{H}} (i-j)},$$

so können wir mit dieser Formel jeden beliebigen Koeffizienten ablesen, wenn wir deg $\mathcal{H} \in O(l)$ Stützstellen $\mathcal{H}(i)$, $i = 0, \ldots, \deg \mathcal{H}$ vorberechnen. $\mathcal{H}(i)$ ist der $t$-te Koeffizient von $F(x, i)^l$, den wir mit Hilfe von Satz 2.5.9 in $\tilde{O}(\sqrt{l})$ Schritten berechnen können. Also benötigen wir $\tilde{O}(l\sqrt{l})$ Operationen und $O(l)$ Speicher, um alle Stützstellen zu ermitteln. Für die Berechnung von $\prod_{j=0, j\neq i}^{deg\ \mathcal{H}} (i-j)$ benötigen wir $O(l)$ Operationen (s. [BGS04], Theorem 1). Damit erhalten wir den gesuchten Koeffizienten in $\tilde{O}(l^{\frac{3}{2}})$ Operationen und $\tilde{O}(l)$ Speicher.

### Weitere Berechnungsmethoden

Im Zuge dieser Arbeit sind noch einige weitere Möglichkeiten zur Berechnung eines Koeffizienten einer bivariaten Polynompotenz in Betracht gezogen worden. Da diese jedoch einen mindestens quadratischen Zeit- und Speicheraufwand in $l$ haben, werden sie hier nur übersichtsartig erwähnt. Es handelt sich dabei um:

- Numerische Anwendungen des *Cauchy'schen Integralsatzes* eines Lifts des Polynoms nach $Z$.
- Diskrete Fouriertransformation, welche Polynom-Multiplikation in Skalarmultiplikation umwandelt.
- Als Spezialfall der Potenzreihenentwicklung wurde die Substitution $x = t^a$, $y = t^b$ zur Isolierung von eines Koeffizienten in $H(t^a, t^b) = \sum_{ij} h_{ij} t^{ia+jb}$ betrachtet und Satz 2.5.9 angewendet.
- Betrachtung verschiedener Interpolationspolynome von $H(x, y)$.

# Kapitel 3

# Modulkurven, Modulsymbole und die Hecke-Algebra

In Kapitel 2 haben wir einen Überblick über die existierenden Algorithmen zur Bestimmung der $\mathbb{F}_{p^n}$-rationalen Punkte der Jacobischen $J_{\mathcal{C}}$ einer Kurve $\mathcal{C}$ über $\mathbb{F}_{p^n}$ gegeben. Für kleine Charakteristik $p$ haben wir $p$-adische und $l$-adische Berechnungsmethoden vorgestellt, für großes $p$ und $n = 1$ gibt es praktikable Algorithmen für Kurven von Geschlecht 1, 2 und 3 basierend auf Kurven mit komplexer Multiplikation oder Berechnung der Hasse-Witt Matrix. Für Kurven von Geschlecht 4 haben wir im hyperelliptischen Fall die $p$-adische Methode von Harvey. Für die viel zahlreicheren nichthyperelliptischen Kurven haben wir jedoch nur die generischen Algorithmen mit bestenfalls subexponentieller Komplexität in der Gruppenordnung, oder die Verwendung der Hasse-Witt Matrix und Auswertung des charakteristischen Polynoms des Frobenius-Endomorphismus mit dem MCT-Algorithmus aus Beispiel 2.5.4 mit einer Gesamtkomplexität von $O(p^{\frac{3}{2}})$.

In Kapitel 4 wollen wir daher einen Algorithmus zur Berechnung der $\mathbb{F}_p$-rationalen Punkte der Jacobischen $J_0(N)$ einer Modulkurve $X_0(N)$ vorstellen. Dieser Algorithmus hat linearen Zeit- und Speicheraufwand in $p$ und eignet sich daher besonders, um nichthyperelliptische Kurven von Geschlecht 4 zu behandeln.

In diesem Kapitel tragen wir alle zum Verständnis von Kapitel 4 nötigen Begriffe und Strukturen zusammen. Wir führen die Modulkurve $X_0(N)$ ein und beschreiben Eigenschaften der Jacobischen Varietät $J_0(N)$ von $X_0(N)$, sowie die Arithmetik der Spitzenformen und der Modulsymbole. Wir geben eine algorithmische Beschreibung der Hecke-Operatoren und zeigen, wie man mit Hilfe der Hecke-Algebra $\mathbb{T}_N$ eine Basis der Spitzenformen $S_2(N)$ berechnen kann.

Dieses Kapitel orientiert sich an dem Artikel von Frey und Müller [FrMü98]. Wir beschränken uns auf Modulformen von Gewicht 2. Für Details und eine allgemeinere Einführung, siehe [Shi94], [Shi73] und [Man72].

## 3.1 Die Modulkurve $X_0(N)$

Sei $\mathrm{SL}_2(\mathbb{Z})$ die Gruppe der ganzzahligen $2 \times 2$ Matrizen mit Determinante 1. $\mathrm{SL}_2(\mathbb{Z})$ wird erzeugt, durch die Matrizen

$$S := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad \text{und} \quad R := \begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}.$$

Für $N \in \mathbb{N}$ definieren wir

$$\Gamma(N) := \left\{ \gamma \in \mathrm{SL}_2(\mathbb{Z}) \ : \ \gamma \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \bmod N \right\},$$

die *Hauptkongruenzuntergruppe der Stufe $N$ zu* $\mathrm{SL}_2(\mathbb{Z})$. $\Gamma(N)$ ist ein Normalteiler von $\mathrm{SL}_2(\mathbb{Z})$. Eine Untergruppe von $\mathrm{SL}_2(\mathbb{Z})$ heißt *Kongruenzuntergruppe zu* $\mathrm{SL}_2(\mathbb{Z})$, falls sie $\Gamma(N)$ für ein $N$ enthält. Wir wollen im folgenden eine spezielle Kongruenzuntergruppe von $\mathrm{SL}_2(\mathbb{Z})$ betrachten:

$$\Gamma_0(N) := \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}) \ : \ c \equiv 0 \bmod N \right\},$$

die *Hecke-Untergruppe* von $\mathrm{SL}_2(\mathbb{Z})$ der *Stufe* $N$. $\mathrm{SL}_2(\mathbb{R})$ und damit $\Gamma_0(N)$ operieren durch die gebrochen lineare Transformation

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot z \mapsto \frac{az+b}{cz+d}$$

auf der komplexen, oberen Halbebene

$$\mathbb{H} := \{ z \in \mathbb{C} \ : \ \mathrm{Im}(z) > 0 \}$$

und durch die selbe Operation auf der erweiterten, oberen Halbebene

$$\mathbb{H}^* := \mathbb{H} \cup \mathbb{P}^1(\mathbb{Q}) = \mathbb{H} \cup \mathbb{Q} \cup \{\infty\},$$

die man erhält, wenn man die Menge der *Spitzen* $\mathbb{Q} \cup \{\infty\}$ an $\mathbb{H}$ adjungiert. Siehe [Shi94], §1.3 – 1.5 für eine eingehende Beschreibung der Topologie von $\mathbb{H}^*$. Eine Umgebungsbasis von $\alpha \in \mathbb{Q}$ ist gegeben durch die Mengen $\{\alpha\} \cup D$, wobei $D$ eine Scheibe in $\mathbb{H}$ darstellt, die die Gerade der reellen Zahlen in $\alpha$ tangiert. Wir bezeichnen die Bahnen von $\Gamma_0(N)$ mit

$$Y_0(N) := \Gamma_0(N) \backslash \mathbb{H} \subseteq \Gamma_0(N) \backslash \mathbb{H}^* =: X_0(N)$$

und nennen $X_0(N)$ die *Modulkurve von* $\Gamma_0(N)$.

$X_0(N)$ (bzw. $Y_0(N)$) erhält eine komplexanalytische Struktur durch die Projektion

$$\pi : \mathbb{H}^* \longrightarrow \Gamma_0(N) \backslash \mathbb{H}^* \quad (\text{ bzw. } \pi : \mathbb{H}^* \longrightarrow \Gamma_0(N) \backslash \mathbb{H} \ ).$$

Den Spitzen muss dabei eine gesonderte Aufmerksamkeit zukommen, siehe [Shi94] Kapitel 1. Mit dieser Struktur ist $Y_0(N)$ eine zusammenhängende eindimensionale komplexe Mannigfaltigkeit. Sie ist eine Riemannsche Fläche mit $\pi$

als Überdeckungsabbildung. $\pi$ ist unverzweigt außerhalb der elliptischen Punkte, welche zu $z = i$ und $z = \zeta_3$ unter der Aktion von $\mathrm{SL}_2(\mathbb{Z})$ konjugiert sind. $X_0(N)$ ist die Kompaktifizierung von $Y_0(N)$, die wir durch Adjungieren der Spitzen von $\Gamma_0(N)$ an $Y_0(N)$ erhalten. Also kann man $X_0(N)$ als eine projektive algebraische Kurve über $\mathbb{C}$ betrachten. Nach SHIMURA ([Shi94], §6.7) ist $X_0(N)$ sogar immer über $\mathbb{Q}$ definiert.

**Beispiel 3.1.1**

1. *Für $N = 1$ hat $X_0(1)$ Geschlecht 0 und daher $X_0(1) \cong \mathbb{P}^1(\mathbb{C})$*

2. *Für $N = 47$ ist $X_0(47)$ eine hyperelliptische Kurve von Geschlecht 4 mit reeller Multiplikation, die gegeben ist durch die affine Gleichung*

$$y^2 = x^{10} + 6x^9 + 11x^8 + 24x^7 + 19x^6 + 16x^5 - 13x^4 - 30x^3 - 38x^2 - 28x - 11.$$

Für eine Funktion $f(z)$ auf $\mathbb{H}$, $k \in \mathbb{N}$ und $\alpha = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{R})$ definieren wir

$$f|_k(\alpha)(z) := \det(\alpha)^{-k/2}(cz + d)^{-k} f(\alpha z).$$

Ist $k = 2$ lassen wir den Zusatz $|_2$ auf der linken Seite der Gleichung weg.

**Definition 3.1.2** *Eine **Modulform der Stufe** $N$ und **Gewicht** $k$ ist eine Funktion $f$ auf $\mathbb{H}$, so dass*

1. *$f$ ist holomorph auf $\mathbb{H}$,*

2. *$f|_k(\alpha)(z) = f(z)$ für alle $z \in \mathbb{H}$ und $\alpha \in \Gamma_0(N)$ und*

3. *$f$ lässt sich holomorph auf die Spitzen erweitern.*

*Verschwindet $f$ zusätzlich in den Spitzen, so sprechen wir von einer **Spitzenform**. Wir bezeichnen den Raum der Modulformen von Gewicht $k$ mit $M_k(N)$ und den Unterraum der Spitzenformen mit $S_k(N)$.*

Siehe [Shi94], §2.1 für eine genaue Definition von Holomorphie und davon, dass $f$ in den Spitzen verschwindet.

Sei $\Omega^1(X_0(N))$ der Raum der holomorphen Differentialformen auf $X_0(N)$. Ein $w \in \Omega^1(X_0(N))$ hat unter der Überdeckungsabbildung $\pi$ lokal die Form $f(z)dz$ mit lokalem Parameter $z$ und $f$ ist eine Spitzenform von Gewicht $k = 2$ und Stufe $N$. Es gilt sogar noch mehr:

**Lemma 3.1.3** *Die Abbildung $f(z) \mapsto 2\pi i \cdot f(z)dz$ induziert einen Isomorphismus zwischen dem Raum der Spitzenformen $S_2(N)$ von Gewicht 2 und dem Raum der holomorphen Differentialformen $\Omega^1(X_0(N))$ von $X_0(N)$. Insbesondere ist die Dimension von $S_2(N)$ als komplexer Vektorraum gleich dem Geschlecht der Modulkurve $X_0(N)$.*

Da $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right) \in \Gamma_0(N)$ gilt nach Definition 3.1.2.2 $f(z+1) = f(z)$ für alle $f \in M_k(N)$. Als periodische Funktion besitzt $f$ also in der Spitze $\infty$ eine Fourier-Entwicklung der Form

$$f(z) = \sum_{n=-\infty}^{\infty} a_n q^n \quad \text{mit } q := e^{2\pi i z}.$$

Die Koeffizienten $a_n$ heißen *Fourier-Koeffizienten* von $f$. Die Holomorphie-Bedingung aus Definition 3.1.2.1 bedingt nun, dass die Koeffizienten $a_n$ für $n < 0$ verschwinden. Ist $f$ eine Spitzenform, so gilt zusätzlich $a_0 = 0$.

Modulformen sind als Fourier-Reihen mit unendlich vielen Koeffizienten gegeben. Es reicht jedoch die Kenntnis von endlich vielen Koeffizienten, um eine Modulform zu identifizieren.

**Satz 3.1.4** *Sei $f \in M_k(N)$ und $\mu := [SL_2(\mathbb{Z}) : \Gamma_0(N)]$. Sind alle Fourier-Koeffizienten $a_n$ von $f$ gleich 0 für $0 \leq n \leq \frac{\mu k}{12}$, so ist $f$ die Nullfunktion.*

**Beweis:** Anwendung von Theorem 8 in [Sch74], S.114 □

## 3.2 Die Hecke-Algebra $\mathbb{T}_N$

$\Gamma_0(N)$ induziert eine *Modulraum*-Struktur auf $Y_0(N)$, die wir im folgenden beschreiben wollen. Sei $\tau \in \mathbb{H}$, dann gibt es eine Bijektion zwischen den Bahnen $\Gamma_0(N)\tau$ und den Isomorphieklassen der Paare $(E, C_N)$ von elliptischen Kurven $E = \mathbb{C}/\Lambda$ mit Periodengitter $\Lambda = \mathbb{Z} + \tau\mathbb{Z}$ und zyklischer Untergruppe $C_N$ der Ordnung $N$ aus $E[N]$. $(E, C_N)$ nennt man dabei eine *erweiterte elliptische Kurve*.

Aus dieser Darstellung kann man ein kanonisches, minimales, eigentliches Modell von $X_0(N)$ über $\mathrm{Spec}(\mathbb{Z})$ konstruieren, welches wir mit $X_0(N)_{\mathbb{Z}}$ bezeichnen. $X_0(N)_{\mathbb{Z}}$ hat gute Reduktion bei allen Primzahlen, die nicht die Stufe $N$ teilen. Wir erhalten also eine $\mathbb{Z}$-Struktur auf $M_k(N)$ und $S_k(N)$ und folgern, dass diese Räume eine Basis von Modul- bzw. Spitzenformen mit ganzzahligen Fourier-Koeffizienten besitzen. Wir erhalten den folgenden Satz (s. [Shi94]).

**Satz 3.2.1 ($q$-expansion-principle)** *Für einen Ring $R$ definieren wir die Tensorprodukte $M_k(N)(R) := M_k(N) \otimes R$ und $S_k(N)(R) := S_k(N) \otimes R$. Dann ist jede Modulform $f \in M_k(N)(R)$* eindeutig *durch ihre Fourier-Koeffizienten bestimmt und diese liegen in $R$.*

Wir nutzen die Modulraum-Struktur auf $X_0(N)_{\mathbb{Z}}$, um eine Gruppe kommutierender Automorphismen von $X_0(N)_{\mathbb{Z}}$ zu definieren. Sei $n$ ein positiver Teiler von $N$ mit $\mathrm{ggT}(n, N/n) = 1$ und $P = (E, C_N)_{\sim}$, dann definieren die Abbildungen

$$w_n(P) := (E/C_n, (C[n] \times C_{N/n})/C_n)_{\sim}$$

Involution von $X_0(N)_{\mathbb{Z}}$, d.h. $w_n^2 = id$. Diese Abbildungen $w_n$ werden *Atkin-Lehner Involutionen* genannt. Nach Ogg [Ogg78] gilt für quadratfreie Stufe $N > 37$

$$\mathrm{Aut}(X_0(N)) = \{w_n : n|N \text{ und } \mathrm{ggT}(n, N/n) = 1\}.$$

**Beispiel 3.2.2** *Die Modulkurve $X_0(53)$ ist eine* ***bielliptische Kurve*** *von Geschlecht $g = 4$, das heißt, es gibt einen Morphismus von Grad zwei von $X_0(53)$ auf eine elliptische Kurve. Der Quotient von $X_0(53)$ mit der Atkin-Lehner-Involution $w_{53}$ ist die elliptische Kurve*

$$y^2 - xy - y = x^3 - x^2.$$

*$X_0(53)$ ist jedoch nicht hyperelliptisch und nach* N. Elkies *wird der Funktionenkörper von $X_0(53)$ dargestellt von $x$, $y$ und einer Wurzel von $f(x, y)$ mit*

$$f(x, y) := x^4 - 7x^3 + 9x^2 - 8x - 11 - (2x^2 + 3x - 11)y.$$

Wir betrachten nun die Abbildung

$$\alpha_n : X_0(nN)_{\mathbb{Z}} \longrightarrow X_0(N)_{\mathbb{Z}} : \quad (E, C_{nN})_{\sim} \mapsto (E, C_N)_{\sim}$$

für $n \in \mathbb{N}$ mit $\mathrm{ggT}(n, N) = 1$. Zusammen mit der Involution $w_n$ erhalten wir eine $\mathbb{Z}$-rationale Korrespondenz

$$\begin{array}{ccc} X_0(nN)_{\mathbb{Z}} & \xrightarrow{w_n} & X_0(nN)_{\mathbb{Z}} \\ \downarrow{\scriptstyle \alpha_n} & & \downarrow{\scriptstyle \alpha_n} \\ X_0(N)_{\mathbb{Z}} & & X_0(N)_{\mathbb{Z}} \end{array}$$

auf $Pic^0(X_0(N))$, der Divisorklassengruppe von Grad 0 von $X_0(N)$:

**Definition 3.2.3** *Für $\mathrm{ggT}(N, n) = 1$ definieren wir*

$$T_n := (\alpha_n)_* \circ w_n \circ (\alpha_n)^*,$$

*den $n$-ten Hecke-Operator.*

Die Erschliessung dieser Abbildung als effizient berechenbarer Algorithmus ist der Mittelpunkt dieser Arbeit. $T_n$ und $w_n$ operieren auf

1. der jacobischen Varietät $J_0(N)$ von $X_0(N)$,
2. dem Raum der Spitzenformen $S_2(N)$,
3. dem Raum der Differentialformen $\Omega^1(X_0(N))$ und dessen Dualraum und
4. der Homologiegruppe $H_1(X_0(N), \mathbb{Z})$. Die Aktion ist dabei verträglich mit der durch die komplexe Konjugation induzierten Aktion auf den Wegen.

**Definition 3.2.4** *Die Hecke-Algebra $\mathbb{T}_N$ von Stufe $N$ ist die $\mathbb{Z}$-Unteralgebra des Endomorphismenrings $\mathrm{End}_{\mathbb{Z}}(\Omega^1(X_0(N)_{\mathbb{Z}})$ die erzeugt wird, durch*

$$w_n \text{ mit } n|N, \ ggT(n, N/n) = 1 \quad \text{und} \quad T_k \text{ mit } ggT(k, N) = 1.$$

Die Hecke-Algebra ist kommutativ (s. [Shi94]), d.h. für Primzahlen $p_1$, $p_2$ wird jeder unter $T_{p_1}$ invariante Unterraum von $M_k(N)$ und $S_k(N)$ unter $T_{p_2}$ wieder in sich abgebildet. Wir können uns auf die Berechnung von $T_p$ mit $p$ prim beschränken:

**Satz 3.2.5** *Die Hecke-Algebra $\mathbb{T}_N$ wird von*

$$w_n \text{ mit } n|N,\ ggT(n, N/n) = 1 \quad \text{und} \quad T_p,\ p \textbf{ prim} \text{ mit } ggT(p, N) = 1$$

*erzeugt.*

**Beweis:** Siehe ([Shi73], Kap. 3) für eine Darstellung von $T_k$ für zerlegbares $k$ durch $T_{p_i}$ mit $p_i$ prim. □

Für $f(z) \in M_k(N)$ und $m \in \mathbb{N}$ definieren wir

$$V_m(f(z)) := f(mz) \quad \text{und} \quad U_m(f(z)) := \frac{1}{m} \sum_{k=0}^{m-1} f\left(\frac{z+k}{m}\right).$$

Damit können wir eine explizite Beschreibung der Aktion von $\mathbb{T}_N$ auf den Modulformen angeben.

**Proposition 3.2.6** *Sei $f \in M_k(N)$ und $n \in \mathbb{N}$ dann gilt*

$$T_n(f(z)) := \left(\sum_{d|n} d^{k-1} V_d \circ U_{n/d}\right)(f(z)).$$

Ist $p$ eine Primzahl, so vereinfacht sich diese Formel zu

$$T_p(f(z)) := p^{k-1} f(pz) + \frac{1}{p} \sum_{k=0}^{p-1} f\left(\frac{z+k}{p}\right).$$

Betrachten wie die Wirkung des Hecke-Operators auf einer in Fourierreihenentwicklung dargestellteter Modulform, so erhalten wir

**Lemma 3.2.7** *Sei $f(z) = \sum_{n=0}^{\infty} a_n q^n \in M_k(N)$ und $p$ eine Primzahl. Dann ist*

$$b_n := a_{pn} + p^{k-1} a_{n/p}$$

*der $n$-te Fourierkoeffizient von $T_p(f(z)) = \sum_{n=0}^{\infty} b_n q^n$, wobei $a_{n/p}$ als null definiert ist, wann immer $p \not| n$.*

Wie zerfällt der Raum der Spitzenformen unter der Wirkung von $\mathbb{T}_N$? Es stellt sich heraus, dass die Antwort auf diese Frage eine Möglichkeit zur Konstruktion von Spitzenformen beinhaltet. Zunächst führen wir einige Begriffe ein.

**Definition 3.2.8** *Eine Modulform $f(z) \in M_k(N)$ ist eine* ***(Hecke-) Eigenfunktion****, falls*

$$T(f(z)) = \lambda_T f(z) \text{ für alle } T \in \mathbb{T}_N.$$

*$\lambda_T$ nennt man dabei* ***(Hecke-)Eigenwert*** *von $T$. Wir bezeichen mit $E_{\lambda_T}$ den* ***$\lambda_T$-Eigenraum***

$$E_{\lambda_T} := \{f(z) \in M_k(N) : T(f(z)) = \lambda_T \cdot f(z)\},$$

*hierbei ist $T \in \mathbb{T}_N$ fest.*

Wir beschränken uns nun auf den Fall von Spitzenformen zu Gewicht $k = 2$ und geben eine Beschreibung der natürlich auftretenden Unterräume von $S_2(N)$. Falls $g \in S_2(M)$ und $M|N$, dann gilt $g(lz) \in S_2(N)$ für $l|(N/M)$. Wir definieren also den Raum der *Altformen* von $S_2(N)$ als

$$S_2^{alt}(N) := \langle g(lz) : g(z) \in S_2(M) \text{ mit } M|N,\ M \neq N,\ l \not| \tfrac{N}{M} \rangle.$$

**Definition 3.2.9** *Das orthogonale Komplement von $S_2^{alt}(N)$ bezüglich des* ***Petersson-Skalarprodukts***

$$\langle f, g \rangle := \int_{\Gamma_o(N)\backslash\mathbb{H}^*} f(z)\overline{g(z)}dxdy \text{ mit } f, g \in S_2(N),\ z = x + iy$$

*bezeichenen wir als den Raum der* ***Neuformen*** *$S_0^{neu}(N)$. Eine Spitzenform $f(z) \in S_0^{neu}(N)$ heißt* ***Neuform****, falls $f(z) = q + \sum_{n\geq 2} a_n q^n$ und $f(z)$ ist eine Hecke-Eigenform.*

Das Petersson-Skalarprodukt ist unabhängig von dem Fundamentalbereich, über den integriert wird und liefert ein hermitesches Skalarprodukt auf dem $\mathbb{C}$-Vektorraum $S_2(N)$. Die Hecke-Operatoren $T_n$ mit $\mathrm{ggT}(N, n) = 1$ bilden $S_0^{neu}(N)$ (bzw. $S_0^{alt}(N)$) in sich selbst ab und sind selbstadjungiert bezüglich des Petersson-Skalarproduktes. $S_2(N)$ zerfällt also in eine direkte Summe aus simultanen Eigenräumen und alle Eigenwerte sind reell. Für $S_2^{neu}(N)$ erhalten wir sogar, dass jeder unter der Hecke-Algebra $\mathbb{T}_N$ simultane Eigenraum $E$ von $S_2^{neu}(N)$ eindimensional ist (*„Multiplicity one theorem“*, s. [AtL70]).

**Satz 3.2.10** *Sei $R$ ein kommutativer Ring. Dann hat die Paarung*

$$\mathbb{T}_N \times S_2(N)(R) \longrightarrow R, \quad a_1(T_n(f)) = \textit{erster Fourier-Koeff. von } T_n(f)$$

*kein linkes und kein rechtes Radikal.*

Aus $S_2(N)(\mathbb{C}) = S_2(N)(\mathbb{Z}) \otimes \mathbb{C}$ folgt nun

**Korollar 3.2.11** *$Hom(\Omega^1(X_0(N)), \mathbb{C})$ ist ein freier $\mathbb{T}_N \otimes \mathbb{C}$-Modul von Rang eins und $\mathbb{T}_N$ ist ein freier $\mathbb{Z}$-Modul von Rang $g(X_0(N))$.*

Zusätzlich erhalten wir eine Möglichkeit zur Konstruktion von Spitzenformen über $R$:

**Proposition 3.2.12** *Sei $\Phi \in Hom(\mathbb{T}_N, R)$. Dann gilt*

$$\sum_{n=1}^{\infty} \Phi(T_n)q^n \in S_2(N)(R).$$

Da $\mathbb{T}_N$ als freier $\mathbb{Z}$-Modul von endlichem Rang auf $S_2(N)$ wirkt, gilt

**Lemma 3.2.13** *Sei $f = q + \sum_{n\geq 2} a_n q^n \in S_2(N)$ eine Hecke-Eigenform und $T \in \mathbb{T}_N$. Dann ist der Eigenwert $\lambda_T$ eine ganzalgebraische, total reelle Zahl und der Körper*

$$K_f := \mathbb{Q}(\lambda_T : T \in \mathbb{T}_N)$$

*ist eine endliche Erweiterung von $\mathbb{Q}$.*

Nach Definition ist $\mathbb{T}_N$ ein Unterring von $\mathrm{End}(J_0(N))$. Ist $N$ quadratfrei, dann gilt nach RIBET

$$\mathbb{T}_N \otimes \mathbb{Q} = \mathrm{End}(J_0(N)) \otimes \mathbb{Q}$$

und falls $N$ eine Primzahl ist, erhalten wir sogar (s. [Maz77])

$$\mathbb{T}_N = \mathrm{End}(J_0(N)).$$

**Bemerkung 3.2.14** Es ist möglich, Hecke- und Atkin-Lehner-Operatoren für Spitzenformen mit Nebentypus und höherem Gewicht zu definieren, s. [AtL70].

## 3.3 Die Arithmetik von $J_0(N)$

Sei $f = q + \sum_{n\geq 2} a_n q^n \in S_2(N)$ eine *Neuform* von Gewicht $k = 2$ und $K_f = \mathbb{Q}(a_1, a_2, \ldots)$ der durch die Adjunktion der Koeffizienten von $f$ erzeugte Körper. Nach Lemma 3.2.13 ist $K_f|\mathbb{Q}$ eine total reelle Körpererweiterung von Grad $d$.

Sei $I_f := \{\sigma_1, \ldots, \sigma_d\}$ die Menge aller paarweise verschiedenen Einbettungen von $K_f$ in $\mathbb{C}$. SHIMURA hat in [Shi94] und [Shi73] zu $f$ eine abelsche Varietät assoziiert:

Sei $a_i$ die Linearform, die einer Spitzenform $f$ ihren $i$-ten Fourierkoeffizienten zuordnet. Die Spitzenform $f$ induziert dann einen Algebra-Homomorphismus

$$\lambda_f : \mathbb{T}_N \otimes \overline{\mathbb{Q}} \longrightarrow \overline{\mathbb{Q}} : \quad T \mapsto a_1(T(f)).$$

Sei $\mathcal{I}_f :=\mathrm{kern}(\lambda_f) \cap \mathbb{T}_N$. Das Bild $\mathcal{I}_f(J_0(N))$ ist ein Untergruppenschema der Jacobischen von $X_0(N)$. Wir definieren

$$A_f := J_0(N)/\mathcal{I}_f(J_0(N)).$$

Dies ist eine abelsche Varietät (s. [Shi94], [Shi73]).

Man sieht leicht, dass $A_f$ über $\mathbb{Q}$ definiert ist und gute Reduktion bei allen Primzahlen hat, die nicht die Stufe $N$ teilen, da selbiges auch für $X_0(N)$ gilt.

$A_f$ läßt sich als komplexer Torus beschreiben, wie in der folgenden Konstruktion nach SHIMURA. Sei $G_{\mathbb{Q}} :=\mathrm{Gal}(\overline{\mathbb{Q}}|\mathbb{Q})$ die absolute Galoisgruppe von $\mathbb{Q}$ und $V$ ein $d$-dimensionaler $G_{\mathbb{Q}}$-invarianter Unterraum von $S_2(N)$ mit Basis $\{f_1, \ldots, f_d\}$. Das Bild der *Abel-Jacobi-Abbildung*

$$\phi : H_1(X_0(N), \mathbb{Z}) \longrightarrow \mathbb{C}^d : \quad \tau \mapsto \left( \int_\tau f_1, \ldots, \int_\tau f_d \right)^t$$

ist ein volles Gitter $\Lambda_V$, d.h. ein diskreter, freier $\mathbb{Z}$-Modul von Rang $2d$ in $\mathbb{C}^d$. Der komplexe Torus $\tilde{A}_v := \mathbb{C}^d/\Lambda_V$ hat die Struktur einer abelschen Varietät (s. [Shi94], [Shi73]).

Für eine Neuform $f$ können wir analog einen $d$-dimensionalen $G_{\mathbb{Q}}$-invarianten Unterraum

$$V_f := \langle f^{\sigma_i} : \sigma_i \in I_f \rangle$$

konstruieren. Die resultierende algebraische Varietät $\tilde{A}_{V_f}$ bezeichnen wir vereinfachend mit $\tilde{A}_f$. SHIMURA hat gezeigt (s. [Shi94])

$$\tilde{A}_f \text{ ist die duale Abelsche Varietät von } A_f \otimes \mathbb{C}.$$

Die wichtige *Eichler-Shimura-Relation* erstellt eine Verbindung zwischen Hecke-Operatoren und den Frobenius-Elementen, die auf $J_0(N)_{|\mathbb{F}_p}$ und ihren verwandten Objekten operieren. Wir zitieren sie für den Tate-Modul $T_l(A_f)$:

**Satz 3.3.1 (Eichler-Shimura-Relation)** *Sei $p \neq l$ eine Primzahl. Für den Frobenius-Endomorphismus $\phi_p$ und den Hecke-Operator $T_p$ gilt*

$$T_p = \phi_p + \phi_p^t$$

*als Endomorphismen des Tate-Moduls $T_l(A_f)$, wobei $\phi_p^t$ der adjungierte Frobenius-Endomorphismus ist.*

Siehe [Shi94] für eine Definition des adjungierten Frobenius-Endomorphismus und für einen Beweis des Satzes.

**Korollar 3.3.2** *Die Koeffizienten der Neuformen erfüllen die Ungleichung*

$$|a_p| \leq 2\sqrt{p}.$$

Sei $C_p$ eine absolut irreduzible projektive glatte Kurve von Geschlecht $g$ über dem endlichen Körper $\mathbb{F}_p$ und $J_{C_p}(\mathbb{F}_p)$ die jacobische Varietät von $C_p$. Falls wir wissen, dass $C_p$ die spezielle Faser der Kurve $C$ ist, deren Jacobische ein Faktor $A_f$ der $J_0(N)$ ist, können wir die Eichler-Shimura-Relation 3.3.1 verwenden, um $\#J_{C_p}(\mathbb{F}_p)$ zu bestimmen: Sei $\chi_{\phi_p}$ das charakteristische Polynom des Frobenius-Endomorphismus $\phi_p$. Sind $e_1, \ldots, e_g$ die Eigenwerte von $T_p$, dann gilt

$$\chi_{\phi_p}(s) = \prod_{i=1}^{g}(1 - e_i s + p s^2)$$

und es ist $(1 - e_i s + p s^2) = (1 - \alpha_i s)(1 - \overline{\alpha_i} s)$ für $1 \leq i \leq g$, wobei $\alpha_i$ die Nullstellen von $\chi_{\phi_p}$ sind.

**Satz 3.3.3** *Sei $\chi_{\phi_p}(s) = 1 + a_1 s + a_2 s^2 + \ldots + a_{2g} s^{2g}$ mit den Nullstellen $\{\alpha_1, \ldots, \alpha_g, \overline{\alpha_1}, \ldots, \overline{\alpha_g}\}$ und $g$ das Geschlecht der Modulkurve $X_0(N)$. Dann gilt*

1. $\chi_{T_p}(p+1) = \chi_{\phi_p}(1)$
2. $|X_0(N)(F_{p^n})| = p^n + 1 - \sum_{i=1}^{g}(\alpha_i^{-n} + \overline{\alpha_i}^{-n})$
3. *Sei $S_n = |X_0(N)(F_{p^n})| - (p^n + 1)$, dann ist*

$$i a_i = S_i a_0 + S_{i-1} a_1 + \ldots + S_1 a_{i-1} \quad \text{für } 1 \leq i \leq g.$$

Für $C_p$ gilt nach Satz 2.2.5:

**Korollar 3.3.4** $\chi_{T_p}(p+1) = \#J_{C_p}(\mathbb{F}_p)$.

Für Beispiele, siehe Abschnitt 4.5 in Kapitel 4.

**Bemerkung 3.3.5** Mit Satz 3.3.3 können wir unter Berechnung von $|X_0(N)(\mathbb{F}_{p^n})|$ für $1 \leq n \leq g$ die Koeffizienten des charakteristischen Polynoms des Frobenius-Endomorphismus ermitteln. Dies ist aber nur für kleine $g$ und $p$ praktikabel.

Ist die Stufe $N \geq 5$ eine Primzahl, so kann die Gruppenordnung $\#J_0(N)$ niemals eine Primzahl sein (s. [Maz77]):

**Satz 3.3.6** *Sei $N \geq 5$ eine Primzahl und $n$ der Zähler des vollständig gekürzten Bruchs $\frac{N-1}{12}$. Dann ist die Torsionsuntergruppe der Mordell-Weil Gruppe von $J_0(N)$ eine zyklische Gruppe der Ordnung $n$.*

Wir verwenden dies, um die mit dem Algorithmus aus Kapitel 4 erstellten Beispiele zu überprüfen. Wir wenden uns jetzt der Berechnung von $\chi_{T_p}(s)$ zu.

## 3.4 Modulsymbole und relative Homologie

Wir beschränken uns nun auf Spitzenformen von Gewicht 2. Sei $H_1(X_0(N), \mathbb{Z})$ die singuläre Homologiegruppe der Riemannfläche $X_0(N)_{\mathbb{R}}$. Dies ist die Abelisierung der Fundamentalgruppe $\pi_1(X_0(N), z)$ mit einem beliebigen Basispunkt $z$. Wir bezeichnen mit $H_1(X_0(N), Spitzen, \mathbb{Z})$ die relative Homologiegruppe von $X_0(N)$ bezüglich der Menge der Spitzen. Man erhält die exakte Sequenz

$$0 \longrightarrow H_1(X_0(N), \mathbb{Z}) \longrightarrow H_1(X_0(N), Spitzen, \mathbb{Z}) \xrightarrow{\delta} \mathbb{Z}^{\nu_\infty} \longrightarrow \mathbb{Z} \longrightarrow 0, \quad (3.1)$$

wobei $\nu_\infty$ die Anzahl der Spitzen von $\Gamma_0(N)$ darstellt. Ein Element von $H_1(X_0(N), Spitzen, \mathbb{Z})$ nennen wir ein *Modulsymbol*. Ein Modulsymbol kann als die Projektion eines Pfades in $\mathbb{H}^*$ dargestellt werden, dessen Anfang und Ende eine Spitze der Modulkurve $X_0(N)$ ist. Da $\mathbb{H}$ einfach zusammenhängend ist, sind alle Pfade $[z_1, z_2]$ zwischen zwei festen Punkten $z_1, z_2 \in \mathbb{H}^*$ äquivalent und ihre Homologieklasse hängt nur von den Endpunkten ab. Wir bezeichnen das Bild von $[z_1, z_2]$ mit $\{z_1, z_2\}$. Die Haupteigenschaften der Modulsymbole sind (s. [Man72]):

**Bemerkung 3.4.1** *Für alle $x, y, z \in \mathbb{H}^*$ gilt*

1. $\{z, z\} = 0$ *und* $\{x, y\} = -\{y, x\}$,
2. $\{x, y\} + \{y, z\} + \{z, x\} = 0$,
3. $\{\alpha x, \alpha y\} = \{x, y\}$ *für alle* $\alpha \in \Gamma_0(N)$

Eine Matrix $\alpha' \in \mathrm{GL}_2(\mathbb{Q})$ operiert dabei auf einem Modulsymbol $\{x, y\}$ durch gebrochen lineare Transformation auf $x$ und $y$.

**Beispiel 3.4.2** Da $\gamma = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right) \in \Gamma_0(N)$, gilt $\{\infty, 0\} = \{\gamma(\infty), \gamma(0)\} = \{\infty, 1\}$. Also folgt $0 = \{\infty, 1\} - \{\infty, 0\} = \{\infty, 1\} + \{0, \infty\} = \{0, \infty\} + \{\infty, 1\} = \{0, 1\}$.

Man kann zeigen, dass die relative Homologiegruppe sich in drei direkte Summanden zerlegen lässt (s. [Mer94])

$$H_1(X_0(N), Spitzen, \mathbb{Z}) = \mathcal{E}is(N) \oplus \mathcal{S}_2(N) \oplus \overline{\mathcal{S}}_2(N),$$

wobei $\mathcal{E}is(N)$ zu den Eisensteinreihen von $\Gamma_0(N)$ und $\mathcal{S}_2(N)$ (bzw. $\overline{\mathcal{S}}_2(N)$) zu den holomorphen (bzw. anti-holomorphen) Spitzenformen korrespondieren.

Sei $J$ die rechte Nebenklassenmenge $\Gamma_0(N)\backslash \mathrm{SL}_2(\mathbb{Z})$. Nach Bemerkung 3.4.1.3 ist

$$\Phi : J \longrightarrow H_1(X_0(N), Spitzen, \mathbb{Z}) : \quad j \mapsto \{\alpha(0), \alpha(\infty)\} \text{ mit } \alpha \in j$$

wohldefiniert.

**Satz 3.4.3** *$H_1(X_0(N), Spitzen, \mathbb{Z})$ ist ein freier $\mathbb{Z}$-Modul von Rang $2g+\nu_\infty-1$, der erzeugt wird durch die Modulsymbole $\{\{\alpha(0), \alpha(\infty)\} : \alpha \in SL_2(\mathbb{Z})\}$.*

Siehe [Man72] für einen Beweis.

Die projektive Gerade über $\mathbb{Z}/N$ ist definiert, durch

$$\mathbb{P}^1(\mathbb{Z}/N) := \{(c, d) \in (\mathbb{Z}/N)^2 : \mathrm{ggT}(c, d, N) = 1\}/\sim$$

mit der Relation

$$(c, d) \sim (c', d') \quad \Longleftrightarrow \quad cd' \equiv c'd \bmod N.$$

Wir bezeichen die Klasse von $(c, d)$ mit $(c : d)$. Man zeigt leicht

$$J \longrightarrow \mathbb{P}^1(\mathbb{Z}/N), \quad j \mapsto (c : d) \bmod N, \text{ mit } j \ni \alpha = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \tag{3.2}$$

ist eine Bijektion zwischen $J$ und der projektiven Geraden $\mathbb{P}^1(\mathbb{Z}/N)$. Dies induziert die $\mathbb{Z}$-lineare Abbildung

$$\delta : \mathbb{Z}[J] \longrightarrow H_1(X_0(N), Spitzen, \mathbb{Z}).$$

Nach Bemerkung 3.4.1.3 gehören die Elemente

$$j + jS \text{ und } j + jR + jR^2$$

zum Kern der Abbildung $\delta$ (Bezeichnungen wie in Abschnitt 3.1). Manin hat gezeigt, dass der Kern von $\delta$ durch diese beiden Elemente erzeugt wird, siehe [Man72]. Wir erhalten also einen Isomorphismus

$$\mathbb{Z}[J]/\langle j + jS, j + jR + jR^2 : j \in J\rangle \;\cong\; H_1(X_0(N), Spitzen, \mathbb{Z}). \tag{3.3}$$

Mit der Bijektion (3.2) haben wir eine explizite Beschreibung von $H_1(X_0(N), Spitzen, \mathbb{Z})$ als $\mathbb{Z}$-Modul, der erzeugt wird durch

$$\mathbb{P}^1(\mathbb{Z}/N) \text{ mit den Relationen: } \; (c : d) + (c : d)S, \; (c : d) + (c : d)R + (c : d)R^2.$$

Die Involution $i : z \to \overline{-z}$ wirkt auf dem Modul durch

$$i((c : d)) = (c : d) \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Definiere

$$\mathcal{M} := \mathbb{Z}[\mathbb{P}^1(\mathbb{Z}/N)]/\langle u + uS, u + uR + uR^2 : u \in \mathbb{P}^1(\mathbb{Z}/N)\rangle.$$

Wir bezeichnen die Klasse von $(c : d) \in \mathbb{P}^1(\mathbb{Z}/N)$ in $\mathcal{M}$ vereinfachend wieder mit $(c : d)$. Die Hecke-Operatoren induzieren eine Aktion von $\mathbb{T}_N$ auf der relativen Homologie. Man sieht leicht, dass für Primzahlen $p \not| N$ gilt

$$T_p(\{c, d\}) = \{pc, pd\} + \sum_{k=0}^{p-1} \left\{ \frac{c+k}{p}, \frac{d+l}{p} \right\}.$$

Wir beschreiben nun diese Aktion auf $\mathcal{M}$ und sammeln dabei Manins und Merels [Mer94] Ergebnisse in

**Satz 3.4.4** *Sei* $M_n := \{M \in \mathrm{Mat}^{2\times 2}(\mathbb{Z}) : \det(M) = n\}$. *Wir definieren für* $M := \left(\begin{smallmatrix} m_1 & m_2 \\ m_3 & m_4 \end{smallmatrix}\right) \in M_n$ *und* $(c : d) \in \mathbb{P}^1(\mathbb{Z}/N)$*:*

$$(c : d)M := \begin{cases} 0 & \textit{falls} \quad (m_1c + m_3d : m_2c + m_4d) \notin \mathbb{P}^1(\mathbb{Z}/N) \\ (m_1c + m_3d : m_2c + m_4d) & \textit{sonst} \end{cases}$$

1. *Für alle* $n \in \mathbb{N}$ *gibt es ein Element*

   $$\Theta_n := \sum u_M M \in \mathbb{Z}[M_n]$$

   *so dass*

   $$\sum u_M((1 : 0)M - (0 : 1)M) = (1 : 0) - (0 : 1) \qquad \textbf{\textit{(Bedingung } $(C_n)$\textit{)}}$$

   *in* $\mathcal{M}$ *gilt.*

2. *Für jedes* $\Theta_n$*, das Bedingung* $(C_n)$ *erfüllt und für* $\mathrm{ggT}(n, N) = 1$ *operiert der* $n$*-te Hecke-Operator* $T_n$ *durch lineare Fortsetzung auf* $\mathcal{M}$ *via*

   $$\mathbb{T}_n((c : d)) = (c : d)\Theta_n = \sum u_M(c : d)M.$$

Für eine effiziente Berechnung benötigen wir eine Abschätzung der Summanden in $\Theta_n$.

**Satz 3.4.5** *Es gibt ein* $\Theta_n$*, so dass* $\#\{M : u_M \neq 0\} \leq O(n \log(n))$.

**Beispiel 3.4.6** Sei $R_n$ die Menge aller ganzzahligen $2 \times 2$ Matrizen der Form $A = \left(\begin{smallmatrix} x & -y \\ y' & x' \end{smallmatrix}\right)$ mit $\det(A) = n$ und eine der folgenden drei Bedingungen ist erfüllt

- $x > |y| > 0$, $x' > |y'| > 0$, und $yy' > 0$,
- $y = 0$ und $|y'| \leq x'/2$,
- $y\prime = 0$ und $|y| \leq x/2$.

Dann erfüllt $\Theta_n = \sum_{M \in \mathbb{R}_n}$ die Bedingung $(C_n)$, siehe [Mer94].

**Beispiel 3.4.7**

$$\Theta_2 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

Für weitere Beispiele, siehe Abschnitt 4.5 in Kapitel 4.

**Bemerkung 3.4.8**

1. $\Theta_n$ hängt weder von der Stufe $N$ noch vom Gewicht $k$ ab.
2. Es gilt die selbe Gleichung, wenn wir $\mathcal{M}$ durch $\mathcal{M}^+$ ersetzen, der zum $+1$-Eigenraum unter der komplexen Konjugation korrespondiert.
3. Zwar ist der Algorithmus über $\mathbb{C}$ gegeben, jedoch hat man nach einem Satz von Manin eine $\mathbb{Z}$-Struktur auf den Symbolen und auf der Hecke-Algebra $\mathbb{T}_N$. Also können wir den Algorithmus über $\mathbb{Z}$ und über endlichen Körpern verwenden.

Nach [Mer94] ist die Aktion der Hecke-Operatoren kompatibel mit der nichtausgearteteten Paarung

$$\mathcal{S}_2(N) \times S_2(N) \longrightarrow \mathbb{C} : \quad \langle \sigma, f(z) \rangle \mapsto \int_\sigma f(z)dz, \tag{3.4}$$

d.h.

$$\langle T_p(\sigma), f \rangle = \langle \sigma, T_p(f) \rangle.$$

Man kann $T_n$ daher durch ganzzahlige Matrizen auf $\mathcal{S}_2(N)$ repräsentieren, denn diese fallen mit der zu dieser Paarung dualen Matrix zusammen. Es ist also möglich, den Raum der Spitzenformen in einfache $\mathbb{T}_N[G_\mathbb{Q}]$-invariante Unterräume zu zerlegen.

Wir wollen nun noch kurz eine wichtige Anwendung der Hecke-Operatoren vorstellen, die jedoch in dieser Arbeit nicht weiter verfolgt wird: Es ist möglich, eine explizite Basis von $S_2(N)(\mathbb{Z})$, bzw. eine Basis von Eigenformen zu berechnen. Die Elemente $f$ dieser Basis sind dann gegeben durch die Fourier-Entwicklung dieser Funktionen. Nach Satz 3.1.4 ist es ausreichend, die ersten $\lfloor \prod_{p|N}(1 + \frac{1}{p}) \rfloor$ Fourier-Koeffizienten zu kennen, um die Spitzenform eindeutig zu identifizieren. Mit $\phi \in \mathrm{Hom}(\mathcal{S}_2(N), \mathbb{Z})$ und $\sigma \in \mathcal{S}_2(N)$ gilt nach Proposition 3.2.12

$$\sum_{n=1}^{\infty} \Phi(T_n(\sigma))q^n \in S_2(N)(\mathbb{Z}) \tag{3.5}$$

Durch variieren von $\sigma$ und $\Phi$ ist es also möglich, eine $\mathbb{Z}$-Basis von $S_2(N)$ zu konstruieren (s. [Mer94]). Setze $\sigma_1 := \sigma$, $\sigma_i + 1 := T_m(\sigma_i)$ mit $\mathrm{ggT}(m, N) = 1$ und $m$ so klein wie möglich. Durch Gaußelimination oder den Wiedemann-Algorithmus identifizieren wir die linear unabhängigen Symbole $\sigma_{i'}$. Sollten diese nicht den ganzen Raum $\mathcal{S}_2(N)$ aufspannen, wählen wir ein anderes Startelement $\sigma'$ oder einen anderen Hecke-Operator $T_{m'}$. Haben wir eine Basis gefunden, so können wir mit Gleichung (3.5) die Fourier-Koeffizienten der korrespondierenden Spitzenformen von $S_2(N)$ finden. In der Praxis reicht es meistens bereits, 2 oder 3 zufällige Elemente $\sigma$ auszuwählen, um eine Basis für $\mathcal{S}_2(N)$ zu finden. Für weitere Informationen, siehe [Bas96] und [FrMü98].

### 3.4.1 Basisdarstellung eines Modulsymbols

Die folgende Basisdarstellung eines Modulsymbols geht zurück auf MANIN. Wir wissen, dass $\Gamma_0(N)$ endlichen Index in $\mathrm{SL}_2(\mathbb{Z})$ hat. Seien $r_0, \ldots, r_m$ ein Repräsentantensystem von $\Gamma_0(N)$ in $\mathrm{Sl}_2(\mathbb{Z})$, so dass

$$\mathrm{SL}_2(\mathbb{Z}) = \Gamma_0(N)r_0 \cup \Gamma_0(N)r_1 \cup \ldots \cup \Gamma_0(N)r_m$$

eine disjunkte Zerlegung von $\mathrm{SL}_2(\mathbb{Z})$ darstellt.

**Beispiel 3.4.9** Für $N$ prim ist

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}, \ldots, \begin{pmatrix} 1 & 0 \\ N-1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

ein vollständiges Repräsentantensystem.

Wir nutzen die Abbildung aus (3.2), um einem Repräsentanten $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ auf das Element $(c : d) \in \mathbb{P}^1(\mathbb{Z}/N)$ abzubilden. Das folgende Verfahren ermöglicht uns, jedes Modulsymbol als Linearkombination von Symbolen der Form $r_i\{0, \infty\}$ zu schreiben. Wegen der Relation $\{\alpha, \beta\} = \{0, \beta\} - \{0, \alpha\}$ genügt es, die Symbole der Form $\{0, b/a\}$ mit $\mathrm{ggT}(a, b) = 1$ zu betrachten. Wir führen eine Kettenbruchzerlegung von $b/a$ durch und behandeln die aufeinanderfolgenden Teilapproximationen mit minimalem Zähler und Nenner

$$\frac{b_{-2}}{a_{-2}} = \frac{0}{1}, \frac{b_{-1}}{a_{-1}} = \frac{1}{0}, \frac{b_0}{a_0} = \frac{b_0}{1}, \ldots, \frac{b_n}{a_n} = \frac{b}{a},$$

wobei die ersten beiden Glieder formal hinzugefügt werden. Dann gilt

$$b_k a_{k-1} - b_{k-1} a_k = (-1)^{k-1}$$

und daraus folgt

$$g_k := \begin{pmatrix} b_k & (-1)^{k-1} b_{k-1} \\ a_k & (-1)^{k-1} a_{k-1} \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}).$$

Damit ist

$$\left\{ \frac{b_{k-1}}{a_{k-1}}, \frac{b_k}{a_k} \right\} = g_k\{0, \infty\} = r_i\{0, \infty\}$$

für $i \in \{0, \ldots, m\}$ von der erwünschten Form.

**Beispiel 3.4.10** Für $N = 11$, betrachte das Modulsymbol $\{0, 4/7\}$. Wir haben

$$\frac{4}{7} = 0 + \frac{1}{1 + \frac{1}{1 + \frac{1}{3}}},$$

also sind die Teilapproximationen von der Form

$$\frac{b_{-2}}{a_{-2}} = \frac{0}{1}, \quad \frac{b_{-1}}{a_{-1}} = \frac{1}{0}, \quad \frac{b_0}{a_0} = \frac{0}{1}, \quad \frac{b_1}{a_1} = \frac{1}{1}, \quad \frac{b_2}{a_2} = \frac{1}{2}, \quad \frac{b_3}{a_3} = \frac{4}{7}.$$

Es gilt $\{0,1\} = 0$ nach Beispiel 3.4.2, also berechnen wir

$$\begin{aligned}
\{0,4/7\} &= \{0,\infty\} + \{\infty,0\} + \{0,1\} + \{1,1/2\} + \{1/2,4/7\} \\
&= \begin{pmatrix} 1 & -1 \\ 2 & -1 \end{pmatrix} \{0,\infty\} + \begin{pmatrix} 4 & 1 \\ 7 & 2 \end{pmatrix} \{0,\infty\} \\
&= \begin{pmatrix} 1 & 0 \\ 9 & 1 \end{pmatrix} \{0,\infty\} + \begin{pmatrix} 1 & 0 \\ 9 & 1 \end{pmatrix} \{0,\infty\} \\
&= 2 \cdot \left[ \begin{pmatrix} 1 & 0 \\ 9 & 1 \end{pmatrix} \{0,\infty\} \right].
\end{aligned}$$

# Kapitel 4

# Ein Algorithmus zur Berechnung von Hecke-Operatoren

Im vorhergehenden Kapitel haben wir den Hecke-Operator $T_n$ vorgestellt und beschrieben, wie man diese Abbildung effektiv berechnen kann, wenn $n$ eine Primzahl ist. Nun stellen wir ein optimiertes Verfahren dazu vor und untersuchen dessen Verhalten in Theorie und Experiment. J. Basmaji hat in [Bas96] ein Verfahren von Merel verschnellert, welches wir in diesem Kapitel aufgreifen und noch einmal verbessern wollen.

Das Verfahren beruht zu einem Großteil auf dem Aufzählen koprimer Tupel, welches wir durch einen Siebvorgang beschleunigen, den wir in Kapitel 5 beschreiben. Der Vorzug von Basmajis Algorithmus, mehrere Hecke-Operatoren $T_{p_1}, \ldots T_{p_n}$ mit Primzahlen $p_1 \equiv p_2 \equiv \ldots \equiv p_n \bmod N$ für Stufe $N$ simultan berechnen zu können, wird hier noch einmal erweitert, indem wir zeigen, wie man die Kongruenzbedingung fallen lassen kann und wie man diese Sequenz $T_{p_1}, \ldots T_{p_n}$ zudem simultan für mehrere Stufen berechnen kann. Wie der Algorithmus von Basmaji bleibt auch dieser voll parallelisierbar, was wir an einigen größeren Beispielen am Ende des Kapitels demonstrieren.

Desweiteren führen wir einige Korrekturen und Klärungen von [Bas96] durch und geben detaillierte Hinweise für eine effiziente Implementierung des hier eingeführten Algorithmus.

## 4.1 Eine bijektive Darstellung von $\overline{S}_p$

J. Basmaji hat in seiner Dissertation [Bas96] einen Algorithmus zur Berechnung von Hecke-Operatoren vorgestellt. Dieser Algorithmus benutzt die Aufzählung der Matrizen aus Definition ([Bas96], 3.2.9.), diese Aufzählung ist jedoch nicht injektiv. Wir beginnen damit, die Definition noch einmal zu zitieren. Dann charakterisieren wir alle Matrizen, die doppelt aufgezählt werden.

**Definition 4.1.1** *Sei $p$ eine ungerade Primzahl. Die Menge $\overline{S}_p$ besteht aus den folgenden Matrizen aus $M_2(\mathbb{Z})$*

1. *Betrachte alle Matrizen der Form* $\begin{pmatrix} p & y \\ 0 & 1 \end{pmatrix}$ *mit* $|y| < \frac{p}{2}$.

2. *Betrachte alle Matrizen der Form* $\begin{pmatrix} 1 & 0 \\ y & p \end{pmatrix}$ *mit* $|y| < \frac{p}{2}$.

3. *Sei* $x, y \in \mathbb{N}$ *mit* $2 \leq x \leq \sqrt{p}$, $0 < y < x$ *mit* $ggT(x, y) = 1$. *Dazu berechne man* $y' := -py^{-1} (mod\ x)$ *und* $x' := \frac{p+yy'}{x}$ *(*$y'$ *sei dabei als größte, negative Zahl gewählt, welche die Kongruenz erfüllt). Ist* $|y'| < x'$, *so betrachten wir die Matrizen*

$$A_k = \begin{pmatrix} x & y \\ y' - kx & x' - ky \end{pmatrix}, \quad B_k = \begin{pmatrix} x & -y \\ -y' + kx & x' - ky \end{pmatrix},$$

$$C_k = \begin{pmatrix} x' - ky & y' - kx \\ y & x \end{pmatrix}, \quad D_k = \begin{pmatrix} x' - ky & -y' + x \\ -y & x \end{pmatrix},$$

$$\text{mit } 0 \leq k \leq r \text{ und } r = \begin{cases} \left\lfloor \frac{x'+y'}{x+y} \right\rfloor & \text{falls } \frac{x'+y'}{x+y} \notin \mathbb{N} \\ \frac{x'+y'}{x+y} - 1 & \text{sonst.} \end{cases}$$

Diese Menge erfüllt die Bedingung $(C_p)$ aus Satz 3.4.4, ihre Aufzählung ist allerdings nicht injektiv. Im folgenden charakterisieren wir alle Matrizen, die doppelt aufgezählt werden.

**Beispiel 4.1.2** Für $p = 11$ erhält man für $x = 3$, $y = 1$ die Matrizen

$$A_0 = \begin{pmatrix} 3 & 1 \\ -2 & 3 \end{pmatrix}, \quad B_0 = \begin{pmatrix} 3 & -1 \\ 2 & 3 \end{pmatrix},$$

$$C_0 = \begin{pmatrix} 3 & -2 \\ 1 & 3 \end{pmatrix}, \quad D_0 = \begin{pmatrix} 3 & 2 \\ -1 & 3 \end{pmatrix}$$

und für $x = 3$, $y = 2$ erhalten wir die Matrizen

$$A_0 = \begin{pmatrix} 3 & 1 \\ -1 & 3 \end{pmatrix}, \quad B_0 = \begin{pmatrix} 3 & -2 \\ 1 & 3 \end{pmatrix},$$

$$C_0 = \begin{pmatrix} 3 & -1 \\ 2 & 3 \end{pmatrix}, \quad D_0 = \begin{pmatrix} 3 & 1 \\ -2 & 3 \end{pmatrix}$$

Die Art, wie die Matrizen hier zusammenfallen, ist charakteristisch für den allgemeinen Fall. Man beachte zusätzlich, dass gilt $x_1x_2 + y_1y_2 = 3 \cdot 3 + 1 \cdot 2 = 11 = p$.

**Lemma 4.1.3** *Seien* $(x_1, y_1), (x_2, y_2) \in \mathbb{N}^2$ *Zahlentupel aus Definition 4.1.1.3 und* $A_k^i, B_k^i, C_k^i, D_k^i$ *die zu den Tupeln* $(x_i, y_i)$, $i = 1, 2$ *gehörigen Matrizen. Dann gilt*

$$x_1x_2 + y_1y_2 = p \quad \Longleftrightarrow \quad A_0^1 = D_0^2,\ B_0^1 = C_0^2,\ C_0^1 = B_0^2,\ D_0^1 = A_0^2.$$

*Dies sind die einzigen Fälle, in denen Matrizen doppelt auftreten. Zu gegebenem Tupel $(x_1, y_1)$ lässt sich also das kritische korrespondierende Tupel beschreiben, durch $x_2 = x_1'$, $y_2 = -y_1'$.*

**Beweis:** Wir vergleichen alle auftretenden Matrizen miteinander und führen dazu einen direkten oder dazu analogen Beweis auf, den wir mit einer römischen Zahl bezeichnen. Es gibt 16 Möglichkeiten, die Matrizen zu kombinieren:

| $\cdot = \cdot$ | $A_k^2$ | $B_k^2$ | $C_k^2$ | $D_k^2$ |
|---|---|---|---|---|
| $A_k^1$ | I | II | III | IV |
| $B_k^1$ | II | I | IV | III |
| $C_k^1$ | III | IV | I | II |
| $D_k^1$ | IV | III | II | I |

**Fall I:** Hier gilt $x_1 = x_2$, $y_1 = y_2$.

**Fall II:** Hier gilt $y_1 = -y_2$, im Widerspruch zu $y_1, y_2 > 0$.

**Fall III:** Es ist

$$\begin{aligned} x_1 &= x_2' - ky_2 && (4.1)\\ y_1 &= y_2' - kx_2 && (4.2)\\ x_2 &= x_1' - ky_1 && (4.3)\\ y_2 &= y_1' - kx_1. && (4.4) \end{aligned}$$

Hier erhalten wir einen Widerspruch aus Gleichung (4.4), da die linke Seite immer und die rechte Seite niemals positiv ist.
**Fall IV:** Es ist

$$\begin{aligned} x_1 &= x_2' - ky_2 && (4.5)\\ y_1 &= -y_2' + kx_2 && (4.6)\\ x_2 &= x_1' - ky_1 && (4.7)\\ y_2 &= -y_1' + kx_1. && (4.8) \end{aligned}$$

Wir zeigen zuerst, dass sich dieses Verhältnis nur bei k=0 einstellen kann. Es folgt aus Gleichung (4.7) und (4.8), sowie aus der Wahl von $y_1'$, $y_2'$ als größte negative Restklassenvertreter

$$\begin{aligned} 0 \geq -y_1 + kx_2 &= y_2' \geq -x_2 && (4.9)\\ 0 \geq -y_2 + kx_1 &= y_1' \geq -x_1. && (4.10) \end{aligned}$$

Ist nun $x_1 \leq x_2$, so gilt $y_1 < x_2$ und aus Gleichung (4.10) folgt dann $k \stackrel{!}{=} 0$. Ist $x_2 \leq x_1$ folgt analog aus (4.10) $k \stackrel{!}{=} 0$.

Damit bleibt nur noch der Fall

$$\begin{aligned} x_1 &= x_2' && (4.11)\\ y_1 &= -y_2' && (4.12)\\ x_2 &= x_1' && (4.13)\\ y_2 &= -y_1'. && (4.14) \end{aligned}$$

Betrachten wir zuerst (4.12). Nach Definition gilt

$$x_1 = x_2' = \frac{p + y_2 y_2'}{x_2} = \frac{p - y_2 y_1}{x_2},$$

wobei der zweite Teil der Gleichung aus (4.14) folgt. Umstellen ergibt

$$x_1 x_2 + y_1 y_2 = p.$$

Selbiges erhält man aus den Gleichungen (4.14) und (4.13). □

**Bemerkung:** Gibt es $x_1, x_2, y_1, y_2 \in \mathbb{N}$ mit $x_1 x_2 + y_1 y_2 = p$, so folgt sofort $\mathrm{ggT}(x_i, y_i) = 1$ für $i = 1, 2$.

Lemma 4.1.3 ermöglicht uns nun eine bijektive Aufzählung der Menge $\overline{S}_p$ aus Definition 4.1.1.

**Definition 4.1.4** *Sei $p$ eine ungerade Primzahl. Die Menge $\overline{S}_p$ besteht aus den folgenden Matrizen aus $M_2(\mathbb{Z})$*

1. *Betrachte alle Matrizen der Form $\begin{pmatrix} p & y \\ 0 & 1 \end{pmatrix}$ mit $|y| < \frac{p}{2}$.*
2. *Betrachte alle Matrizen der Form $\begin{pmatrix} 1 & 0 \\ y & p \end{pmatrix}$ mit $|y| < \frac{p}{2}$.*
3. *Sei $s := \sqrt{p}, x, y \in \mathbb{N}$ mit $2 \leq x \leq s$, $0 < y < x$ mit $ggT(x, y) = 1$. Dazu berechne man $y' := -py^{-1} (mod\ x)$ und $x' := \frac{p+yy'}{x}$ ($y'$ sei dabei als größte, negative Zahl gewählt, welche die Kongruenz erfüllt). Ist $|y'| < x'$ so betrachten wir die Matrizen*

$$A_k = \begin{pmatrix} x & y \\ y' - kx & x' - ky \end{pmatrix}, \quad B_k = \begin{pmatrix} x & -y \\ -y' + kx & x' - ky \end{pmatrix},$$

$$C_k = \begin{pmatrix} x' - ky & y' - kx \\ y & x \end{pmatrix}, \quad D_k = \begin{pmatrix} x' - ky & -y' + x \\ -y & x \end{pmatrix},$$

*mit $j \leq k \leq r$ und* $r = \begin{cases} \left\lfloor \frac{x'+y'}{x+y} \right\rfloor & \text{falls } \frac{x'+y'}{x+y} \notin \mathbb{N} \\ \frac{x'+y'}{x+y} - 1 & \text{sonst.} \end{cases}$

*Dabei gilt* $j = \begin{cases} 0 & \text{falls } x < x' \quad \text{oder} \quad x = x' \ \wedge \ y < -y' \\ 1 & \text{falls } x > x' \quad \text{oder} \quad x = x' \ \wedge \ y \geq -y'. \end{cases}$

*Im Fall $x = x' \wedge y = -y'$ zählen zusätzlich die Matrizen $A_0$ und $B_0$ zu $\overline{S}_p$.*

**Bemerkung:** Der Fall $x = x' \wedge y = -y'$ tritt nur auf, falls $p$ sich als Summe von zwei Quadraten darstellen lässt. Dies ist äquivalent zu $p \equiv 1 \bmod 4$, siehe [Neu92].1.1.

In dem nachfolgenden Algorithmus werden die Matrizen aus Definition 4.1.4.3 für die Werte

$$\begin{aligned} x &= 2 \quad y = 1, \\ x &= 3 \quad y = 1, \\ x &= 3 \quad y = 2 \end{aligned}$$

gesondert behandelt. Das folgende Lemma garantiert, dass wir dabei keine doppelten Matrizen zu erwarten haben.

**Lemma 4.1.5** *Sei $p \in \mathbb{N}$ eine Primzahl und $p > 25$. Dann gibt es keine im Sinn von Lemma 4.1.3 doppelten Matrizen in der Aufzählung von Definition 4.1.4.3 für die Werte*

$$\begin{aligned} x &= 2 \quad y = 1, \\ x &= 3 \quad y = 1, \\ x &= 3 \quad y = 2. \end{aligned}$$

**Beweis:** Für $x = 3$, $y = 2$ findet kann man nach Lemma 4.1.3 nur eine doppelte Matrix konstruieren, falls man ein Paar $x_2, y_2 \in \mathbb{N}$ hat, mit $\sqrt{p} \geq x_2 > y_2$, $\mathrm{ggT}(x_2, y_2) = 1$ und

$$3x_2 + 2y_2 = p.$$

Nun schätzen wir $y_2$ nach unten ab, durch

$$y_2 = \frac{p - 3x_2}{2} \geq \frac{p - 3\sqrt{p}}{2} = \sqrt{p}\left(\frac{\sqrt{p} - 3}{2}\right)$$

und erhalten daher einen Widerspruch, sobald $p > 25$. Die anderen beiden Fälle folgen analog. □

## 4.2 Der Algorithmus

Ist $x, y$ ein mögliches Zahlenpaar aus Definition 4.1.4.3 und $x \leq \tilde{x} \in \mathbb{N}$ und $y \leq \tilde{y} \in \mathbb{N}$ mit $x \equiv \tilde{x} \bmod N$ und $y \equiv \tilde{y} \bmod N$, dann erzeugen beide Zahlenpaare mit der Aufzählung aus Definition 4.1.4.3 modulo $N$ die selben Matrixfolgen. Es gibt also zur Matrix $\tilde{A}_0$ aus dem Paar $(\tilde{x}, \tilde{y})$ ein $k$ mit $\tilde{A}_0 \equiv A_k \bmod N$ usw. Man kann die Häufigkeit der auftretenden Matrizen modulo $N$ in einem Vektor $V = (V_k)_{k=0,\ldots,N-1}$ zählen. Dies wird im Algorithmus [Bas96].3.3.3 genutzt, um die Operationen auf den Modulsymbolen zu reduzieren. Diesen Algorithmus behandeln wir nun mit der Aufzählung aus Definition 4.1.4.

In Kapitel 5 zeigen wir, dass die Verwendung eines Siebes zur Erstellung der nötigen teilerfremden Tupel zeitlich effizienter ist, als ein Verfahren, das auf dem euklidischen Algorithmus basiert. Dies wird in dem nachfolgenden Algorithmus berücksichtigt. Anschliessend sind noch einige Bemerkungen zu Verbesserungen gegenüber Basmajis Algorithmus in [Bas96].3.3.3. aufgeführt.

**Algorithmus 4.2.1** Berechnung des Heckeoperators $T_p$

**Eingabe:** Stufe $N$, Primzahl $p > 25$, $(u : v) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$
**Ausgabe:** $\sum_{M \in \overline{S}_p} (u : v)M$

1. Betrachtung der Matrizen aus Definition 4.1.4.1 und 4.1.4.2:
   Berechne $q, r \in \mathbb{N}_0$ mit $p = qN + r$ und

$$q \sum_{k=0}^{N-1} \left[ (u : v) \begin{pmatrix} p & k \\ 0 & 1 \end{pmatrix} + (u : v) \begin{pmatrix} 1 & 0 \\ k & p \end{pmatrix} \right].$$

   Addiere die restlichen $r$ Matrizen

$$\sum_{k=0}^{r} \left[ (u : v) \begin{pmatrix} p & k \\ 0 & 1 \end{pmatrix} + (u : v) \begin{pmatrix} 1 & 0 \\ k & p \end{pmatrix} \right].$$

2. Selbes Vorgehen wie unter Punkt 1 für die Matrizenfolgen $A_k$, $B_k$, $C_k$ und $D_k$ aus Definition 4.1.4.3 für die Tupel

$$\begin{aligned} x = 2 \quad & y = 1, \\ x = 3 \quad & y = 1, \\ x = 3 \quad & y = 2. \end{aligned}$$

3. Nun zählen wir alle kongruenten Matrizen modulo $N$:
   Für alle $\tilde{x}$ mit $4 \leq \tilde{x} < N + 4$
   und alle $\tilde{y}$ mit $1 \leq \tilde{y} \leq N$:

   (a) Ist $\mathrm{ggT}(\tilde{x}, \tilde{y}, N) = 1$, dann
   Durchlaufe $x$ mit $\tilde{x} \leq x \leq \sqrt{p}$ und $x \equiv \tilde{x} \bmod N$

   (b) Erstelle mittels Sieben (s. Kapitel 5) eine Liste $L_{\tilde{y},x}$ aller zu $x$ teilerfremden Zahlen zwischen $\tilde{y}$ und $x$.

   (c) Durchlaufe $y$ mit $\tilde{y} \leq y < x$ und $y \equiv \tilde{y} \bmod N$ und prüfe $y \in L_{\tilde{y},x}$. Ist $(x, y)$ das erste gefundene teilerfremde Paar, setze $\hat{x} = x$ und $\hat{y} = y$. Findet man auf diese Weise kein teilerfremdes Paar, so setze $x$ auf $x + N$ und gehe zurück zu 3.($b$).

   (d) Berechne die Matrix

$$S_p(\hat{x}, \hat{y}) = \begin{pmatrix} \hat{x} & \hat{y} \\ \hat{y}' & \hat{x}' \end{pmatrix} \quad \text{mit} \quad \begin{aligned} \hat{y}' &\equiv -p\hat{y}^{-1} \bmod \hat{x}, \\ \hat{x}' &= \tfrac{p + \hat{y}\hat{y}'}{\hat{x}}. \end{aligned}$$

   $\hat{y}'$ ist dabei als grösster negativer Repräsentant seiner Restklasse gewählt.

(e) Ist $|\hat{y}'| < \hat{x}'$, berechne $q, s \in \mathbb{N}_0$ mit $r = qN+s$ und $r$ wie in Definition 4.1.4.3. Setze in dem Häufigkeitsvektor $V_k = q$ für $k = 0, \ldots N-1$, dann erhöhe $V_k$ um eins für die Einträge $k = 1, \ldots s-1$. Prüfe anhand der Kriterien aus Definition 4.1.4.3, ob $S_p(\hat{x}, \hat{y})$ als „doppelt“ gilt, sonst erhöhe $V_0$ ebenfalls um eins.

(f) Erstelle mittels Sieben die noch nachfolgenden teilerfremden Tupel $(x, y)$. Berechne dazu wie unter 3.(d) die Matrix $M_p(x, y)$ und bestimme $k \in \{0, \ldots, N-1\}$ mit

$$M_p(x,y) \equiv \begin{pmatrix} \hat{x} & \hat{y} \\ \hat{y}' - k\hat{x} & \hat{x}' - k\hat{y} \end{pmatrix} \bmod N$$

Ist $|y| < x$, berechne wie in 3.(e) die Anzahl $r$ der Matrizen in der Matrixfolge und füge diese dem Häufigkeitsvektor $V$ beginnend bei $k$ dazu. Berücksichtige eventuell auftretende „Doppelte“ bei $V_0$.

(g) Sind für festes $\tilde{x}$ und $\tilde{y}$ die Schleifen für $x$ und $y$ durchlaufen, so berechne für $(\hat{x}, \hat{y})$ die $4N$ Matrixfolgen $A_k$, $B_k$, $C_k$ und $D_k$ aus Definition 4.1.4.3 für $k = 0, \ldots, N-1$ und wende diese auf das Element $(u : v)$ an. In dem Vektor $V$ kann man aus dem $k$-ten Eintrag ablesen, wie oft das Bild unter der jeweiligen Matrixoperation auftritt. Man berechnet also

$$\sum_{k=0}^{N-1} V_k \cdot [(u:v)A_k + (u:v)B_k + (u:v)C_k + (u:v)D_k]$$

und addiert diese Summe zu den vorher erstellten Symbolen.

---

**Bemerkung 4.2.2** Nachfolgend sind einige Erklärungen aufgeführt, die die Implementation von Algorithmus 4.2.1 erleichtern sollen. Auch auf Verbesserungen gegenüber Algorithmus [Bas96].3.3.3 wird hier hingewiesen.

**zu Punkt 4.2.1.2** Man beachte, dass die Startmatrizen für $x = 3$ davon abhängen, ob $p \equiv 1 \bmod 3$ oder $p \equiv 2 \bmod 3$.

**zu Punkt 4.2.1.3.(a)** Ist $ggT(\tilde{x}, \tilde{y}, N) \neq 1$, so gibt es für diesen Durchlauf keine teilerfremden Tupel.

**zu Punkt 4.2.1.3.(d)** In dem Ausdruck „$\hat{y}' = -p\hat{y}^{-1} \bmod \hat{x}$“ gibt es einen Druckfehler in Algorithmus [Bas96].3.3.3, wie man bereits bei einfachen Beispielen sehen kann. Auch kann man in den Bemerkungen zu Algorithmus [Bas96].3.3.3 den Eindruck erhalten, die Matrix $S_p(\hat{x}, \hat{y})$ würde modulo $N$ nicht von $p$ abhängen. Hier kann man sich ebenfalls mit einfachen Beispielen vom Gegenteil überzeugen. Dies ist vor allem für die simultane Berechnung mehrerer modulo $N$ kongruenter Primzahlen $p$ wichtig, vgl. den nachfolgenden Abschnitt.

**zu Punkt 4.2.1.3.(e)** Anders als in Algorithmus [Bas96].3.3.3. beschrieben, benötigt man nur einen Vektor $V_k$, anstatt vier, um das Vorkommen der Matrizen modulo $N$ zu zählen.

**zu Punkt 4.2.1.3.(f)** Für den ersten Durchlauf kann man noch das zu $\hat{x}$ erstellte Sieb benutzen.

**zu Punkt 4.2.1.3.(f)** Auch $M_p(x, y)$ hängt hier modulo $N$ von $p$ ab, was man bei der simultanen Berechnung mehrerer modulo $N$ kongruenter Primzahlen berücksichtigen muss.

**zu Punkt 4.2.1.3.(f)** Ist $\tilde{x}$ oder $\tilde{y}$ modulo $N$ invertierbar, so kann man $k$ berechnen aus $(\hat{y}' - y')/\hat{x} \bmod N$, bzw. $(\hat{x}' - x')/\hat{y} \bmod N$. Hat man $k$ erst gefunden, so ist es überflüssig, die nachfolgenden Matrizen explizit aufzuzählen, da diese modulo $N$ die gleiche Folge bilden, wie die von $S_p(\hat{x}, \hat{y})$. Daher kann man die Anzahl der auftretenden Matrizen sofort in $V$ abspeichern.

### 4.2.1 Simultanes Berechnen mehrerer Symbole

Algorithmus 4.2.1 kann simultan mehrere Symbole $(u_1, v_1), \ldots, (u_t, v_t)$ berechnen, wenn man in Punkt 4.2.1.1, Punkt 4.2.1.2 und 4.2.1.3.(g) eine Schleife hinzufügt, die über die Liste der Symbole iteriert. Verwendet man eine Basis des Modulraums, so erhält man auf diese Weise die Matrix für den Hecke-Operator $T_p$.

### 4.2.2 Simultanes Berechnen mehrerer beliebiger Hecke-Operatoren

In [Bas96] stellt J. Basmaji einen Algorithmus vor, mit dem man Serien $T_{p_1}, T_{p_2}, \ldots, T_{p_l}$ von Hecke-Operatoren mit $p_1 \equiv p_2 \equiv \ldots \equiv p_l \bmod N$ berechnen kann. Wir zeigen nun, wie man diese Kongruenzbedingung mit nur marginalen Zeiteinbußen weglassen kann. Das Verfahren lässt sich leicht auf Algorithmus 4.2.1 übertragen, indem man die folgenden Schritte hinzufügt:

1. In Punkt 4.2.1.1 eine Schleife über alle Primzahlen $p_1, \ldots, p_l$. Man beachte, dass man nur einmal alle resultierenden Symbole aufzusummieren hat, falls die Kongruenzbedingung für die $p_i$ erfüllt sind. Dies ist die einzige Stelle, an der man von dieser Kongruenzbedingung profitiert.

   Das selbe gilt für die in Punkt 4.2.1.2 anfallenden Schritte.

2. In 4.2.1.3.(a) durchlaufe alle $x$ mit $\tilde{x} \leq x \leq \max_i \sqrt{p_i}$ und $x = \tilde{x} \bmod N$.

3. Das Erstellen der teilerfremden Tupel ist von $p_1, \ldots, p_l$ unabhängig, die Matrizen $S_p(\hat{x}, \hat{y})$ und $M_p(x, y)$ jedoch nicht, abgesehen von der Inversenbildung $\hat{y}^{-1} \bmod \hat{x}$ . Füge also eine Schleife über 4.2.1.3.(d) bis 4.2.1.3.(e) hinzu, die über $p_1, \ldots p_l$ iteriert. Der Fall $x > p_i$ kann dabei nun auftreten und muss gesondert abgefangen werden. Selbes Vorgehen für 4.2.1.3.(f). Man benötigt also $l$ Häufigkeitsvektoren $(V_{ij})_{i=0,\ldots N-1,\ j=1,\ldots l}$.

4. Füge eine Schleife über 4.2.1.3.(g) hinzu, die durch $p_1, \ldots, p_l$ iteriert. Man beachte, dass die Matrixfolgen von der jeweiligen Primzahl $p_i$ abhängig sind.

### 4.2.3 Simultanes Berechnen mehrerer Stufen

Der Vorteil der Verwendung eines Siebes zur Erstellung der teilerfremden Tupel liegt nicht nur in der höheren zeitlichen Effizienz, man bekommt zudem noch das komplette Intervall aller teilerfremden Tupel $L_{\tilde{y},x}$ zu gegebenem $x$ und Untergrenze $\tilde{y}$. Damit hängt die Erstellung dieser Tupel nicht mehr von der Stufe ab und wir sind in der Lage, Algorithmus 4.2.1 so zu erweitern, dass man einen oder mehrere Heckeoperatoren für eine beliebige endliche Folge $N_1, N_2, \ldots, N_n$ von Stufen gleichzeitig berechnen kann. Dafür sind folgende Erweiterungen hinzuzufügen:

1. In Punkt 4.2.1.1 eine Schleife über alle Stufen $N_1, N_2, \ldots, N_n$. Man beachte, dass man nur einmal alle resultierenden Symbole aufzusummieren hat. Das selbe gilt für die in Punkt 4.2.1.2 anfallenden Schritte.

2. Abschnitt 4.2.1.3 wird nun über alle $\tilde{x}$ mit $4 \leq \tilde{x} < \alpha + 4$ und alle $\tilde{y}$ mit $1 \leq \tilde{y} \leq \alpha$ iteriert. Hierbei gilt $\alpha = \max_i N_i$.

3. In Punkt 4.2.1.3.(a) wird nun über $x$ in Schritten von $\mathrm{ggT}(N_1, \ldots, N_n)$ iteriert, anstatt in Schritten von $N$. Der Test $\mathrm{ggT}(\tilde{x}, \tilde{y}, N) = 1$ fällt weg.

4. In Punkt 4.2.1.3.(b) erstelle nur dann ein Sieb $L_{\tilde{y},x}$, falls es ein $N_i$ gibt, mit

   (a) $x \equiv \tilde{x} \bmod N_i$,

   (b) $\tilde{x} < N_i + 4 \quad \text{und} \quad \tilde{y} \leq N_i$,

   (c) $\mathrm{ggT}(\tilde{x}, \tilde{y}, N_i)$=1.

5. In Punkt 4.2.1.3.(c) iteriere $y$ in Schritten von $\mathrm{ggT}(N_1, \ldots, N_n)$. Ist ein teilerfremdes Paar gefunden, so führe die Punkte 4.2.1.3.(d) und 4.2.1.3.(e) für alle $N_i$ mit $y \equiv \tilde{y} \bmod N_i$ aus. Beachte, dass die Matrix $S_{p,N_i}(\hat{x}, \hat{y}) := S_p(\hat{x}, \hat{y})$ nur einmal für alle diese $N_i$ erstellt werden muss. Den Sonderfall, dass zu einem $N$ kein teilerfremdes Paar gefunden wurde, kann man durch die Erstellung eines gesonderten Siebes für $x + N$ behandeln (bzw. $x + 2N$, $x + 3N \ldots$ bei wiederholtem Misserfolg). Die in dieser Phase zu berechnenden Siebe sind minimal zeitaufwändig. Jedes $N_i$ erhält einen eigenen Häufigkeitsvektor $V_i$.

6. Für jedes teilerfremde Tupel $(x, y)$ in Punkt 4.2.1.3.(f) wird nun ebenso verfahren. Für alle $N_i$ mit

   (a) $x \equiv \tilde{x} \bmod N_i$,

   (b) $y \equiv \tilde{y} \bmod N_i$,

   (c) $\tilde{x} < N_i + 4 \quad \text{und} \quad \tilde{y} \leq N_i$,

(d) $\mathrm{ggT}(\tilde{x}, \tilde{y}, N_i)=1$.

(e) $x \geq (S_{p,N_i}(\hat{x}, \hat{y}))_{11}$ (gemeint ist das zu $N_i$ gehörige $\hat{x}$),

erstelle die Matrix $M_p(x, y)$, bestimme $k_{N_i} \in \{0, \ldots, N_i - 1\}$ und trage die Häufigkeit der Matrizen in die Vektoren $V_i$ ein. Beachte, dass $M_p(x, y)$ nur einmal für alle diese $N_i$ berechnet werden muss, ebenso wie die Gesamtlänge $r$ der Matrizenfolge.

7. Füge eine Schleife über $N_1, N_2, \ldots, N_n$ dem Punkt 4.2.1.3.(g) hinzu. Beachte, dass die Matrixfolgen zu $(\hat{x}, \hat{y})$ nun von dem jeweiligen $N_i$ abhängen.

Da die meisten Modulräume unterschiedliche Symbole als Basis haben, empfiehlt es sich, zu jeder Stufe $N$ eine Liste der entsprechenden Basiselemente abzuspeichern und in dem Programm das Symbol $(u : v)$ iterativ durch diese Basiselemente zu ersetzen. So kann man die Matrizen der zugehörigen Hecke-Operatoren auch für mehrere Stufen simultan berechnen.

### 4.2.4 Simultanes Berechnen mehrerer Stufen und mehrerer Hecke-Operatoren

Mit der Verallgemeinerung von Algorithmus 4.2.1 auf die simultane Berechnung beliebieger Hecke-Operatoren in Abschnitt 4.2.2 ist es problemlos möglich, mehrere Argumente beider Eingabeparameter zu berechnen, indem man Abschnitt 4.2.2 und 4.2.3 gleichzeitig anwendet.

### 4.2.5 Parallelisierbarkeit von Algorithmus 4.2.1

Da nach jedem Durchlauf die erhaltenen Modulsymbole zu den restlichen addiert werden müssen, erhält man an dieser Schnittstelle eine bequeme Möglichkeit, Algorithmus 4.2.1 in $d$ unabhängige Prozesse $\phi_1, \ldots, \phi_d$ aufzuteilen, indem man folgende Schritte hinzufügt:

1. Prozess $\phi_1$ führt zusätzlich zu den im folgenden aufgeführten Aufgaben die Punkte 4.2.1.1 und 4.2.1.2 aus. Dies führt zu einem vernachlässigbaren Mehraufwand für $\phi_1$.

2. Prozess $\phi_i$ führt Punkteblock 4.2.1.3 aus, falls $\tilde{x} \equiv i - 1 \bmod d$ ist. Sollte $N < d$ gelten, so kann man zudem die Prozesse über Kongruenzklassen von $\tilde{y}$ modulo $d$ aufteilen.

3. Für das Gesamtergebnis muss man nun noch die erhaltenen Teilsummen von Modulsymbolen aus den einzelnen Prozessen aufsummieren.

Praktische Tests haben ergeben, dass diese Aufteilung den Aufwand in nahezu gleiche Anteile zerlegt.

### 4.2.6 Modulsymbolreduktion mit Magma

Nachdem die Matrizen in Punkt 4.2.1.1. und Punkt 4.2.1.3.(g) auf das Symbol $(u : v)$ angewendet wurden, müssen die Modulsymbole noch aufsummiert und in Basisdarstellung gebracht werden. Dazu verwenden wir das Computeralgebrasystem Magma. Mit dem Befehl

```
M:=ModularSymbols(N,2,+1);
```

definieren wir in Magma einen Unterraum $M := \mathcal{E}(N) \oplus \mathcal{S}_2(N)$ der Modulsymbole von Gewicht 2 und Stufe $N$ der zu den Eisensteinreihen von $\Gamma_0(N)$ und den holomorphen Spitzenformen $S_2(N)$ korrespondiert. Eigentlich sind wir nur an $\mathcal{S}_2(N)$ interessiert, jedoch bietet die aktuelle Magma-Version keine Möglichkeit, den Raum weiter einzuschränken. Zudem bietet dieser Raum die Möglichkeit einer Korrektheitsüberprüfung von Algorithmus 4.2.1, da jedes Bild von $(u : v)$ unter einer Matrix von Magma in einer Basis von $\mathcal{E}(N) \oplus \mathcal{S}_2(N)$ dargestellt wird, die endgültige Summe aller dieser Bilder muss aber ein Element von $\mathcal{S}_2(N)$ sein, falls $(u : v) \in \mathcal{S}_2(N)$.

Da Algorithmus 4.2.1 mit Maninsymbolen rechnet, müssen wir die Bilder $(r, s)$ unter den Matrizen für Magma in Modulsymbole überführen. Dies geschieht mit dem Befehl (vgl. [Magma] und Abschnitt 3.4)

```
ConvertFromManinSymbol(M,<1,[r,s]>);
```

Hat man die Symbole einmal in dieser Form deklariert, ermöglicht Magma eine natürliche Additionsarithmetik. Das Ergebnis wird dann automatisch in der Basis von $M$ dargestellt. Magma hat auch eine eigene Funktion, die Matrix für den Hecke-Operator zur Primzahl $p$ zu berechnen:

```
HeckeOperator(M,p);
```

Diese Funktion eignet sich jedoch nur für sehr kleine Primzahlen und Stufen, da Speicheraufwand und Zeitbedarf sehr schnell wachsen.

## 4.3 Komplexität des Algorithmus

Wir wollen nun das theoretische Laufzeitverhalten von Algorithmus 4.2.1 untersuchen und dem Algorithmus [Bas96].3.3.3 von Basmaji gegenüberstellen. Algorithmus 4.2.1 kann man im wesentlichen in vier verschiedene Aufgaben aufteilen:

1. Das Auffinden der teilerfremden Tupel mittels Sieben in Punkt 4.2.1.3.(b), (c) und (f).

2. Das Erstellen der Matrizen $S_p(\hat{x}, \hat{y})$ in Punkt 4.2.1.3.(d) und das Erstellen der Matrizen $M_p(x, y)$ in Punkt 4.2.1.3.(f).

3. Das Anwenden der Matrizen auf ein Symbol in Punkt 4.2.1.1 und Punkt 4.2.1.3.(g).

4. Das Auffinden von $k$ in Punkt 4.2.1.3.(f).

Für diese Aufgaben wollen wir im folgenden den theoretischem Aufwand abschätzen. Zunächst benötigen wir eine Abschätzung über die Häufigkeit eines koprimen Zahlenpaares.

**Lemma 4.3.1** *Für zwei zufällig gewählte Zahlen $x, y \in \mathbb{Z}$ liegt die Wahrscheinlichkeit für* $\mathrm{ggT}(x,y) = 1$ *bei*

$$\frac{6}{\pi^2} \approx 0,6079.$$

**Beweis:** Die Wahrscheinlichkeit, dass eine Zahl durch eine Primzahl $p$ teilbar ist, liegt bei $1/p$. Die Wahrscheinlichkeit, dass zwei Zahlen durch diese Primzahl $p$ teilbar ist, liegt also bei $1/p^2$ und daher ist die Wahrscheinlichkeit, dass mindestens eine der beiden Zahlen nicht durch $p$ geteilt wird gleich $1 - 1/p^2$.

Ist $P$ die Menge aller Primzahlen, so erhalten wir die Wahrscheinlichkeit für ein koprimes Tupel als Grenzwert des Produktes über alle Primzahlen

$$\prod_{p \in P} \left(1 - \frac{1}{p^2}\right) = \left(\prod_{p \in P} \frac{1}{1 - p^{-2}}\right)^{-1} = \frac{1}{\zeta(2)} = \frac{6}{\pi^2}.$$

$\zeta(s) = \sum_{n=1}^{\infty} 1/n^s$ ist dabei die Riemannsche-Zetafunktion. □

### 4.3.1 Zeitkomplexität des Algorithmus

Abschnitt [Bas96].3.3.4 kann man entnehmen, dass der Algorithmus [Bas96].3.3.3 für feste Stufe $N$ und Primzahl $p$ maximal

$$\frac{1}{2}(N^2 - 1)\left(\frac{\lfloor\sqrt{p}\rfloor}{N} + 1\right)\left(\frac{\lfloor\sqrt{p}\rfloor}{N} + 2\right) \tag{4.15}$$

ggT-Berechnungen durchführen muss. Dies liegt daran, dass man für Algorithmus [Bas96].3.3.3 ein spezielles Restsystem $\mathcal{R}_{\mathcal{N}}$ von $(\mathbb{Z}/N\mathbb{Z})^2$ durchlaufen muss. Für ein festes $(\tilde{x}, \tilde{y}) \in \mathcal{R}_{\mathcal{N}}$ bestimmt man alle Paare

$$(\tilde{x}, \tilde{y}) \equiv (x, y) \bmod N \quad \text{mit} \quad \begin{array}{l} \tilde{x} \le x \le \sqrt{p} \text{ und } \tilde{y} \le y \le x, \\ \mathrm{ggT}(x, y) = 1. \end{array}$$

Das heißt, man muss für ein Element $(\tilde{x}, \tilde{y}) \in \mathcal{R}_{\mathcal{N}}$ maximal

$$\sum_{k=1}^{\lfloor \frac{\lfloor\sqrt{p}\rfloor - \tilde{x}}{N} \rfloor + 1} k = \frac{1}{2}\left(\frac{\lfloor\sqrt{p}\rfloor - \tilde{x}}{N} + 1\right)\left(\frac{\lfloor\sqrt{p}\rfloor - \tilde{x}}{N} + 2\right)$$

ggT-Berechnungen durchführen. Die Anzahl der Paare $(\tilde{x}, \tilde{y}) \in \mathcal{R}_{\mathcal{N}}$, die durchlaufen werden müssen, ist nach oben abzuschätzen, durch

$$N^2 - \#\left\{(\tilde{x}, \tilde{y}) \in \mathcal{R}_N \ : \ \mathrm{ggT}(\tilde{x}, \tilde{y}, N) \neq 1\right\} \le N^2 - 1$$

und wir erhalten Formel (4.15).

Bei Algorithmus 4.2.1 muss man das selbe Restsystem $\mathcal{R}_N$ in Punkt 3 durchlaufen, jedoch werden die teilerfremden Tupel $(x, y)$ nicht durch explizites Berechnen des ggT bestimmt, sondern durch Aussieben aller zu $x$ teilerfremden Zahlen, die kleiner sind als $x$. Ein Siebvorgang benötigt $O(x)$ Additionen in $\mathbb{Z}$ und berechnet alle relevanten $y$ in diesem Schritt. Es gibt maximal $\lfloor \frac{\lfloor\sqrt{p}\rfloor - \tilde{x}}{N} \rfloor + 1$ Siebvorgänge und daher werden alle zu $\tilde{x}$ gehörigen Tupel mit

$$O\left(\sum_{k=1}^{\lfloor \frac{\lfloor\sqrt{p}\rfloor - \tilde{x}}{N} \rfloor + 1} k\right) \subset O\left(\frac{1}{2}\left(\frac{\lfloor\sqrt{p}\rfloor - \tilde{x}}{N} + 1\right)\left(\frac{\lfloor\sqrt{p}\rfloor - \tilde{x}}{N} + 2\right)\right)$$

Additionen abgearbeitet. Da das Sieben nur von $\tilde{x}$ abhängt, kann man die Gesamtzahl aller nötigen Additionen in $\mathbb{Z}$ abschätzen durch

$$O\left(\frac{N}{2}\left(\frac{\lfloor\sqrt{p}\rfloor}{N} + 1\right)\left(\frac{\lfloor\sqrt{p}\rfloor}{N} + 2\right)\right). \tag{4.16}$$

Wir haben also in Algorithmus 4.2.1 zeitaufwändige *ggT-Berechnungen* gegen *Additionen in* $\mathbb{Z}$ ersetzt.

Kommen wir nun zur Erstellung der Matrizen $S_p(\hat{x}, \hat{y})$ und $M_p(x, y)$ in Algorithmus 4.2.1. Dies entspricht der Berechnung der Matrizen $S$ und $M$ in Algorithmus [Bas96].3.3.3.5-6. Der grösste Aufwand bei der Erstellung dieser Matrizen ist die Berechnung von

$$\hat{y}' = -p\hat{y}^{-1} \bmod \hat{x}.$$

Mit dem erweiterten euklidischen Algorithmus kann man feststellen, ob zwei Zahlen teilerfremd sind und gegebenenfalls gleichzeitig das modulare Inverse der Zahlen ausrechnen. Aus Abschnitt 5.3 geht jedoch hervor, dass es zeitlich für Algorithmus [Bas96].3.3.3 günstiger ist, diese beiden Aufgaben zu trennen. Also müssen Algorithmus 4.2.1 und Algorithmus [Bas96].3.3.3 die selbe Anzahl an Inversenberechnungen durchführen. Die theoretische Wahrscheinlichkeit, dass zwei zufällig gewählte Zahlen zueinander teilerfremd sind, beträgt ungefähr $0,6079$ (s. Lemma 4.3.1). Betrachtet man die Formel (4.15), so sind also knapp 61% davon als Inversenberechnung durchzuführen.

Genauso oft muss man zu gegebenem $M_p(x, y)$ in Punkt 4.2.1.3.(f) und [Bas96].3.3.3.6. das entsprechende $k$ suchen. Ist $\tilde{x}$ oder $\tilde{y}$ modulo $N$ invertierbar, so kann man $k$ damit durch eine Multiplikation in $\mathbb{Z}/N\mathbb{Z}$ bestimmen, wie in Bemerkung 4.2.2 beschrieben. Ist weder $\tilde{x}$ noch $\tilde{y}$ modulo $N$ invertierbar, so bleibt nur noch die Möglichkeit $k$ durch Ausprobieren zu suchen, in welchem Fall man maximal $N$ Additionen $\mathbb{Z}/N\mathbb{Z}$ durchzuführen hat. Wir benutzen wieder $0,6079$ als Wahrscheinlichkeit für zwei zufällige teilerfremde Zahlen und erhalten $(1 - 0,6079)^2 \approx 0,1537$ als Wahrscheinlichkeit dafür, dass man $k$ wie im letzteren Fall durch Brute-Force suchen muss. Man beachte, dass die

Zusammensetzung von $N$ einen entscheidenden Einfluss auf diesen Abschnitt hat.

Es bleibt noch die Operation der erstellten Matrizen auf dem Modulsymbol auszuführen. Die Erstellung der Matrizen wird durch $\tilde{x}$ und $\tilde{y}$ gesteuert und da man insgesamt vier Matrixfolgen modulo $N$ auf das Symbol anwenden muss, hat man maximal

$$4N(N^2-1)$$

Operationen dieser Art auszuführen.

Für kleine Stufe $N$ und große Primzahl $p$ wird der theoretische Aufwand von Algorithmus 4.2.1 durch die Inversenbildung bei der Erstellung der Matrizen $S$ und $M$, sowie das Auffinden von $k$ dominiert. Man hat im wesentlichen

$$O(Np)$$

solcher Operationen durchzuführen. Gegenüber Algorithmus [Bas96].3.3.3. haben wir nun große Zeiteinsparungen bei der Erstellung der teilerfremden Tupel und dem Auffinden von $k$.

### 4.3.2 Speicherkomplexität des Algorithmus

Der Algorithmus [Bas96].3.3.3. von Basmaji benötigt im wesentlichen $4N \log p$ Bit zum Speichern der Häufigkeitsvektoren $V$. Da in Algorithmus 4.2.1 die teilerfremden Tupel durch Sieben ermittelt werden, benötigen wir zusätzlich $O(\sqrt{p})$ Bit Speicher für das Sieb. Um die Teiler der zu siebenden Zahl möglichst schnell zu ermitteln, werden für alle Zahlen $x \leq \sqrt{p}$ ihre Teiler in einer Tabelle abgespeichert, wie in Abschnitt 5.2.1 beschrieben. Sei $\pi(x)$ die Anzahl aller Primzahlen kleiner oder gleich $x$. Dann benötigt man $\pi(x)$ Bit für den Eintrag von $x$ in die Teilertabelle. Nach einem Satz von TSCHEBYSCHEW gilt asymptotisch

$$\pi(x) \sim \frac{x}{\log x}.$$

Das heißt, wir benötigen asymptotisch $O(p \log \sqrt{p})$ Bit für die Teilertabelle. Die Grösse der Teilertabelle lässt sich apriori genau abzählen. Siehe Abschnitt 4.4 für den konkreten Steigungsverlauf des Speicherbedarfs.

### 4.3.3 Simultanes Berechnen mehrerer Modulsymbole

Die Erstellung der Matrizen und ihre Anwendung auf ein Modulsymbol sind zwei voneinander unabhängige Aufgabenteile. Für die Berechnung von $h$ Symbolen hat man also insgesamt einen Mehraufwand von $O(hN^2(N-1))$ für die Anwendung der Matrixfolgen modulo $N$ in Abhängigkeit von $\tilde{x}$ und $\tilde{y}$. Der zusätzliche Speicher ist $O(h \log N)$ Bit zum Speichern der Ausgangs- und Zielsymbole.

### 4.3.4 Simultanes Berechnen mehrerer Hecke-Operatoren

Der große Vorzug von Algorithmus 4.2.1 gegenüber dem Algorithmus [Bas96].3.3.3. von Basmaji liegt in der effizienten Berechnung von Serien von Hecke-Operatoren $T_{p_1}, \ldots, T_{p_l}$, zu Primzahlen $p_1, \ldots, p_l$, die in keiner Weise eine Relation zueinander haben müssen. Dies eröffnet uns auch die Möglichkeit, diese Hecke-Operatoren gleichzeitig zu mehreren Stufen zu berechnen.

Die Auffindung teilerfremder Tupel und die modulare Inversenbildung zur Erstellung der Matrizen $S_{p_i}(\hat{x}, \hat{y})$ und $M_{p_i}(x, y)$ sind von $p_i$ unabhängig und müssen insgesamt nur einmal berechnet werden. Man beachte, dass diese Inversenbildung neben dem Auffinden von $k$ in Punkt 4.2.1.3.(f) der zeitintensivste Arbeitsteil ist. Dies wird sich in der praktischen Auswertung in Abschnitt 4.4 bestätigen.

Der zusätzliche Speicher beläuft sich auf $O(l \log N)$ Bit zum Abspeichern der Matrizen $S_{p_i}(\hat{x}, \hat{y})$ und $M_{p_i}(x, y)$ und und $O(lN \log p)$ Bit zum Speichern der Häufigkeitsvektoren $V$.

### 4.3.5 Simultanes Berechnen mehrerer Stufen

Ein weiterer Vorzug, den Algorithmus 4.2.1 gegenüber Algorithmus [Bas96]3.3.3. bietet, ist die simultane Berechnung mehrerer Stufen in einem Arbeitsgang. Wir wollen nun eine *worst-case* und eine *best-case* Abschätzung für den zeitlichen Aufwand des Algorithmus machen. Seien $N_1, \ldots, N_n$ die eingegebenen Stufen. Für gegebenes $\tilde{x}$ in Punkt 4.2.1.3 des Algorithmus spart man einen Siebvorgang, sobald man in 4.2.1.3.(a) oder 4.2.1.3.(f) ein $x$ und und zwei Indizes $i, j$ gefunden hat, mit

$$x \equiv \tilde{x} \bmod N_i \quad \text{und} \quad x \equiv \tilde{x} \bmod N_j.$$

Im schlechtesten Fall gilt $\mathrm{ggT}(N_i, N_j) = 1$ und damit $\mathrm{kgV}(N_i, N_j) = N_i N_j$. Das heißt, zu gegebenem $\tilde{x}$ gibt es minimal

$$\frac{\sqrt{p}}{\alpha^2} \tag{4.17}$$

gemeinsame Benutzungen eines Siebes, wenn $\alpha$ das Maximum aller $N_1, \ldots, N_n$ ist. Damit sparen wir für jedes $\tilde{x}$

$$\frac{p}{\alpha^2}$$

Additionen beim Sieben und Inversionen modulo $\hat{x}$ bzw. $x$ bei der Erstellung der Matrizen $S_p(\hat{x}, \hat{y})$ und $M_p(x, y)$ (mit Faktor $0,6079$ für die Wahrscheinlichkeit für ein teilerfremdes Tupel, s. Lemma 4.3.1). Es gibt $n(n-1)/2$ Möglichkeiten, $i$ und $j$ zu kombinieren und mindestens $\delta := \min_i\{N_i\}$ Möglichkeiten für $\tilde{x}$, also spart man insgesamt

$$O\left(\frac{\delta n(n-1)}{2\alpha^2} p\right)$$

Additionen bzw. Inversionen. Wir lassen dabei außer Acht, dass auch mehr als zwei Stufen zusammen fallen können.

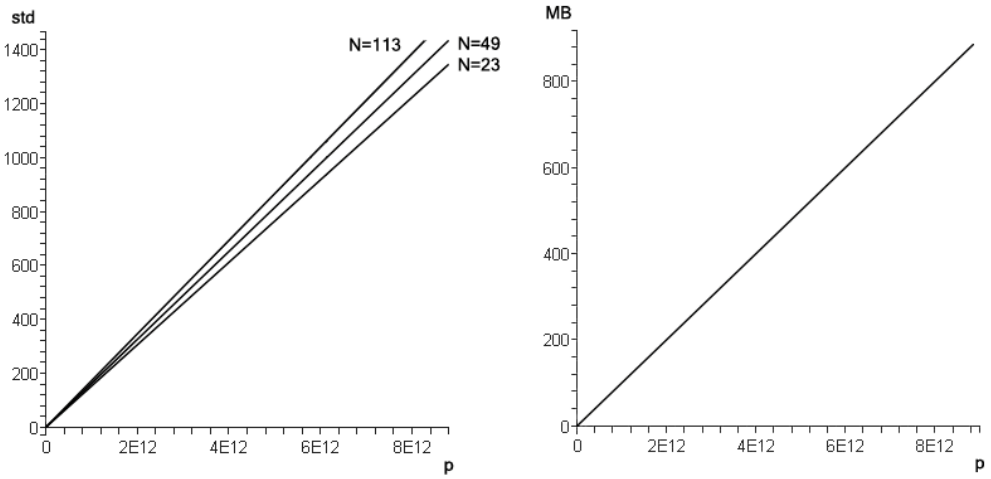


Abbildung 4.1: Zeitbedarf in Stunden und Speicherbedarf in MB in Abhängigkeit von der Primzahl $p$ für die Stufen $N = 23, 47, 113$.

Basierend auf diesen Ergebnissen wollen wir nun Parameter extrapolieren, die ein günstiges Laufverhalten des Algorithmus bewirken. Eine Möglichkeit ist, die Stufen so zu wählen, dass möglichst viele davon einen großen Faktor $\beta$ enthalten. Dieser tritt dann als Faktor in Gleichung (4.17) wieder auf. Ist $\beta \sim \delta$, so spart man mindestens

$$O\left(\frac{n(n-1)}{2\alpha}p\right)$$

Inversionen und Additionen. Die zweite Möglichkeit ist, $n \sim \alpha$ zu wählen. Dann spart man

$$O\left(\frac{\delta(\alpha-1)}{2\alpha}p\right)$$

Inversionen und Additionen. Beide Ansätze sind kumulativ, man kann aber nicht beide gleichzeitig maximieren.

Man benötigt $O(n \log \alpha)$ Bit zusätzlichen Speicher zum Speichern der Matrizen $S_{p,N_i}(\hat{x}, \hat{y})$ und $O(n\alpha \log p)$ Bit zum Speichern der Häufigkeitsvektoren $V$.

## 4.4 Experimentelle Ergebnisse

Die Implementation von Algorithmus 4.2.1 spiegelt die linearen Komplexitätseigenschaften des Verfahrens genau wieder. Das Programm wurde in `C++` unter Linux unter Verwendung des Compilers `gcc 4.1.1` mit dem Flag -O3 erstellt. Die Langzahlarithmetik wurde mit `gmp 4.2.1` durchgeführt. Zur Reduktion der in Punkt 4.2.1.1, Punkt 4.2.1.2 und Punkt 4.2.1.3.(g) auftretenden Modulsymbole wurde das Computeralgebrasystem `Magma` in der Version `V2.11-8` verwendet. Die Rechnungen fanden auf einem 2300 MHz Athlon Prozessor mit 512 KB Cache und 4 GB RAM statt.

Abbildung 4.1 zeigt die interpolierten Werte zu den Stufen $N = 23, 47$ und 113. Es zeigte sich sehr deutlich, dass die Rechenzeit linear von irreduziblen Stufen $N$ und der Primzahl $p$ des Hecke-Operators $T_p$ abhängig ist. Lässt man auch reduzible Zahlen als Stufen zu, so wird der lineare Charakter der Komplexität in $N$ leicht dadurch beeinträchtigt, dass man in Punkt 4.2.1.3.(f) den Faktor $k$ öfter durch Brute-Force suchen muss. Erwartungsgemäß benötigt man für die Berechnung mehr Zeit, wenn $N$ kleine Primfaktoren enthält. Tests, in denen $N$ aus dem doppelten eine Primzahl bestand, benötigten 5% mehr Zeit als proportional große irreduzible Stufen. War $N$ jedoch das Quadrat einer Primzahl, so war der Zeitbedarf praktisch unverändert. Basmaji vergleicht in [Bas96].3.4 seinen Algorithmus mit dem von Merel ([Bas96].3.1.1.) und betont, dass sein Algorithmus [Bas96].3.3.3. vor allem effizent für große Primzahl $p$ gegenüber der Stufe $N$ ist. Dies gilt auch für Algorithmus 4.2.1.

Der Hauptanteil der Berechnung liegt in der Erstellung der Matrizen. Die Vorbereitung der Teilertabelle und die abschließende Reduktion der Modulsymbole nahmen einen vernachlässigbaren Anteil an der Rechenzeit in Anspruch.

Der Speicherbedarf steigt linear und stellt keine große Einschränkung dar. Im Jahr der Fertigstellung dieser Arbeit kostete 1 GB RAM etwa 50 Euro. Abbildung 4.1 können wir entnehmen, dass man mit 900 MB Speicher eine 43-Bit Primzahl für Stufen $N$ unter 100 innerhalb von 60 Tagen berechnen kann. Das Verfahren aus Abschnitt 4.2.5 ermöglicht praktikabel diesen Arbeitsaufwand zu parallelisieren. Siehe Abschnitt 4.5 für ein Beispiel.

### 4.4.1 Simultanes Rechnen

Ist $g$ das Geschlecht der zugrundeliegenden Modulkurve, so benötigt man nur $g$ Modulsymbole, um eine Matrix für den zu berechnenden Hecke-Operator zu erstellen. Der Mehraufwand für das simultane Berechnen mehrerer Modulsymbole steigt extrem langsam und ist daher für niedrige Stufe vernachlässigbar.

Die simultane Berechnung mehrerer Hecke-Operatoren hat eine konstante Zeitersparnis. Getestet wurden variable Serien bis zu 100 Hecke-Operatoren bis 32 Bit zu Kurven mit Stufen zwischen 20 und 150. Man spart etwa 55% der Gesamtzeit gegenüber einer sequentiellen Berechnung der selben Hecke-Operatoren. Dies zeigt auch, dass der Aufwand durch das Sieben etwa die Hälfte des Gesamtaufwands ausmacht, da dies nur einmal für alle beteiligten Hecke-Operatoren durchgeführt wird.

Auch die Zeitersparnis bei der simultanen Berechnung mehrerer Stufen ist konstant bei 20%. Getestet wurden Serien von 50 Modulkurven mit zufälliger Stufe zwischen 5 und 100 zu Hecke-Operatoren mit 32-Bit.

## 4.5 Beispiele

Die nachfolgenden Beispiele sind mit einigen weiteren Beispielen noch einmal im Anhang aufgeführt.

**Beispiel 4.5.1** Wir betrachten die Modulkurve $X_0(47)$, eine hyperelliptische Kurve von Geschlecht $g = 4$ mit reeller Multiplikation. Diese ist gegeben, durch

die affine Gleichung

$$y^2 = x^{10} + 6x^9 + 11x^8 + 24x^7 + 19x^6 + 16x^5 - 13x^4 - 30x^3 - 38x^2 - 28x - 11.$$

Entnommen ist diese Kurve aus [Web97], Anhang B, Tabelle 8. Eine Basis für den zu den Spitzenformen gehörigen Raum der Modulsymbole $\mathcal{S}_2(47)$ ist gegeben durch die Modulsymbole

$$m_1 := \{-1/28, 0\} \quad m_2 := \{-1/35, 0\} \quad m_3 := \{-1/31, 0\} \quad m_4 := \{-1/23, 0\}.$$

Die Matrix des Hecke-Operators $T_p$ zu der 43 Bit großen Primzahl $p = 8796093027097$ ist

$$B_{T_p} = \begin{pmatrix} -2136278 & -426764 & 119260 & -1187176 \\ -927236 & -186678 & -1094060 & -2616896 \\ 1306436 & 379200 & -236254 & 1520824 \\ -165818 & -1201254 & -142036 & -186678 \end{pmatrix}.$$

Die Matrixdarstellung nach der Reduktion durch `Magma` war invariant unter dem zu den holomorphen Spitzenformen gehörigen Modulraum (siehe Abschnitt 4.2.6). Bei Primzahlen dieser Grösse ist damit ein inkorrektes Ergebnis praktisch ausgeschlossen. Die Zusammenfassung der Modulsymbole hat 7 Minuten und 10 Sekunden und 3 MB Speicher benötigt.

Das charakteristische Polynom $\chi_{T_p}$ von $B_{T_p}$ ist

$$\begin{aligned} \chi_{T_p}(x) &= x^4 + 2745888x^3 - 1835697173864x^2 \\ &\quad -7420787822715538048x - 318168613214847820753854 4. \end{aligned}$$

Die Ordnung von $J_0(47)_p(\mathbb{F}_p)$ ist

$$\begin{aligned} \chi_{T_p}(p+1) &= 59863125885736215240413939656509260530354654256622 08 \\ &= 2^8 \cdot 23 \cdot 1153 \\ &\quad \cdot 8817841377546554952406461472274192048972354 47. \end{aligned}$$

Da die Stufe von $X_0(47)$ eine Primzahl ist, besagt Satz 3.3.6, dass die Torsionsgruppe von $J_0(47)$ über $\mathbb{Q}$ die Ordnung 23 hat. Diesen Faktor finden wir in $\#J_0(47)_p(\mathbb{F}_p)$ wieder, was die Korrektheit der Rechnung zusätzlich unterstützt. Der letzte Faktor von $\#J_0(47)_p(\mathbb{F}_p)$ ist eine 150-Bit Primzahl.

Dieses Beispiel wurde in 10 Tagen mit sechs Prozessen auf drei Athlon Dualcore 4600+, also sechs Prozessoren mit jeweils 2300 MHz, berechnet. Alle Prozesse sind in etwa gleichzeitig geendet. Der erforderliche Speicheraufwand betrug 900 MB. $\chi_{T_p}(p+1)$ ist eine Zahl mit 172 Bit. Obwohl dies kein kryptografisch relevantes Beispiel ist, so zeigt sich doch, dass man innerhalb von 10 Tagen in diesen Bereich vordringen kann.

Betrachten wir ein weiteres Beispiel für diese Kurve, den Heckeoperator $T_{p'}$ zur 40 Bit großen Primzahl $p' = 1099511634451$. Die resultierende Matrix ist

$$B_{T_{p'}} = \begin{pmatrix} -772224 & -1354016 & -1967644 & -1429168 \\ 564280 & 755640 & 1203712 & 1029864 \\ -538476 & 25804 & -596116 & 150304 \\ 651204 & 859322 & 1315310 & 755640 \end{pmatrix}.$$

Auch diese Darstellungsmatrix war wieder invariant in den holomorphen Modulsymbolen unter der Reduktion in `Magma`, die Korrektheit ist also praktisch sicher. Die Reduktion hat `Magma` in 5 Minuten und 41 Sekunden durchgeführt und dabei 3 MB Speicher belegt.

Das charakteristische Polynom $\chi_{T_{p'}}$ von $B_{T_{p'}}$ ist

$$\begin{aligned}\chi_{T_{p'}}(x) &= x^4 - 142940x^3 - 1515164718304x^2 \\ &\quad -56211709293679488x + 27223726644522392367872\end{aligned}$$

Die Ordnung $J_0(47)_{p'}(\mathbb{F}_{p'})$ ist

$$\begin{aligned}\chi_{T_{p'}}(p'+1) &= 14615014828249227406830817845977431519874427957 76 \\ &= 2^8 \cdot 23 \cdot 248216963794993671991012531351518877715258627.\end{aligned}$$

Auch hier finden wir wieder die Ordnung der Torsionsgruppe von $J_0(47)$ wieder. Bei dem letzten Faktor handelt es sich um eine 147-Bit große Primzahl.

Rechner und Versuchsaufbau waren wie oben. Der Längste der sechs Prozesse hat 31 Stunden und 34 Minuten, der Kürzeste 25 Stunden und 56 Minuten benötigt. Dies sind die Zeiten im einzelnen:

| Prozess | $\phi_1$ | $\phi_2$ | $\phi_3$ | $\phi_4$ | $\phi_5$ | $\phi_6$ |
|---|---|---|---|---|---|---|
| std:min | 31:22 | 31:15 | 26:37 | 31:34 | 25:56 | 26:18 |

Der Prozess $\phi_1$ hat zusätzlich die Punkte 4.2.1.1 und 4.2.1.2 von Algorithmus 4.2.1 ausgeführt.

Wir führen nun noch ein Beispiel einer nichthyperelliptischen Modulkurve auf.

**Beispiel 4.5.2** Die Modulkurve $X_0(53)$ aus Beispiel 3.2.2 ist eine *bielliptische Kurve* von Geschlecht $g = 4$. Wir wissen, es gibt einen Morphismus von Grad zwei von $X_0(53)$ auf eine elliptische Kurve. Der Quotient von $X_0(53)$ mit der Atkin-Lehner-Involution $w_{53}$ ist die elliptische Kurve

$$y^2 - xy - y = x^3 - x^2.$$

$X_0(53)$ ist jedoch nicht hyperelliptisch und nach N. Elkies wird der Funktionenkörper von $X_0(53)$ dargestellt von $x$, $y$ und einer Wurzel von $f(x, y)$ mit

$$f(x, y) := x^4 - 7x^3 + 9x^2 - 8x - 11 - (2x^2 + 3x - 11)y.$$

Eine Basis für den zu den Spitzenformen gehörigen Raum der Modulsymbole $\mathcal{S}_2(53)$ ist gegeben durch die Modulsymbole

$$m_1 := \{-1/21, 0\} \quad m_2 := \{-1/13, 0\} \quad m_3 := \{-1/35, 0\} \quad m_4 := \{-1/26, 0\}.$$

Die Matrix des Hecke-Operators $T_p$ zu der 43 Bit großen Primzahl $p = 8796093025123$ ist

$$A_{T_p} = \begin{pmatrix} -5524207 & 0 & 0 & 0 \\ 3191271 & -452979 & -639597 & 1311314 \\ -4525359 & -3869702 & -4322681 & -5181016 \\ 4478495 & 1295254 & 1934851/2 & 2137529 \end{pmatrix}.$$

Auch hier wurde die Korrektheit der Matrix durch die Invarianz in $\mathcal{S}_2(53)$ unter der Reduktion in `Magma` praktisch sichergestellt.

Das charakteristische Polynom $\chi_{T_p}$ von $A_{T_p}$ ist

$$\begin{aligned}\chi_{T_p}(x) &= x^4 + 8162338x^3 + 7162266210134x^2 \\ &\quad -44291241380068260924x - 18503885240506467856404701.\end{aligned}$$

Die Ordnung von $J_0(53)_p(\mathbb{F}_p)$ ist

$$\begin{aligned}\chi_{T_p}(p+1) &= 59863162694454335876243931149411019855175878979093 95 \\ &= 3^5 \cdot 5 \cdot 13^2 \cdot 839 \cdot 5227 \cdot 565567 \cdot 36197936417 \\ &\quad \cdot 324723964356328319527111.\end{aligned}$$

Nach Satz 3.3.6 hat die Torsionsgruppe von $J_0(53)$ die Ordnung 13. Diese Zahl tritt auch wieder in $\#J_0(53)_p(\mathbb{F}_p)$ auf und wir erhalten eine zusätzliche Bestätigung für die Korrektheit der Berechnung.

Rechner und Versuchsaufbau waren wie in Beispiel 4.5.1. Die folgende Tabelle zeigt die benötigten Zeiten der Prozesse. Der Prozess $\phi_1$ hat zusatzlich die Punkte 4.2.1.1 und 4.2.1.2 von Algorithmus 4.2.1 ausgeführt. Der längste Prozess $\phi_2$ benötigte 11 Tage und 4 Stunden.

| Prozess | $\phi_1$ | $\phi_2$ | $\phi_3$ | $\phi_4$ | $\phi_5$ | $\phi_6$ |
|---|---|---|---|---|---|---|
| std:min | 263:10 | 268:10 | 231:25 | 259:16 | 213:40 | 219:46 |

Der erforderliche Speicheraufwand betrug 900MB pro Prozess. Die Teilertabelle benötigte davon 885 MB und hätte als statisches Speicherelement zentral für alle Prozesse ausgelagert werden können. In Anbetracht des vorhandenen Speichers wurde davon jedoch abgesehen, da eine Konkurrenz der Prozesse um diese Daten eine Verzögerung der Rechnung mit sich gebracht hätte.

Wir führen noch ein weiteres Beispiel für diese Kurve auf. Die Matrix des Hecke-Operators $T_{p'}$ zu der 43 Bit großen Primzahl $p' = 8796093030211$ ist

$$A_{T_{p'}} = \begin{pmatrix} 829037 & 0 & 0 & 0 \\ -1174937 & -3973191 & 425005 & 2452354 \\ -1978511 & -752334 & -4725525 & -3204688 \\ -1199356 & 801172 & 376167/2 & -2370847 \end{pmatrix}.$$

Das charakteristische Polynom $\chi_{T_{p'}}$ von $A_{T_{p'}}$ ist

$$\begin{aligned}\chi_{T_{p'}}(x) &= x^4 + 10240526x^3 + 29179398120126x^2 \\ &\quad +8021305065282605760x - 33012454281031794147865253.\end{aligned}$$

Die Ordnung von $J_0(53)_{p'}(\mathbb{F}_{p'})$ ist

$$\begin{aligned}\chi_{T_{p'}}(p'+1) &= 59863176976394671641013978400705875590823384418172 75 \\ &= 3 \cdot 5^2 \cdot 13 \cdot 41 \cdot 5849 \cdot 489061 \\ &\quad \cdot 52351199397385217820899871508643367281\end{aligned}$$

Auch hier tritt wieder der Faktor 13 aus Satz 3.3.6 auf. Der letzte Faktor ist eine 125-Bit Zahl. Versuchsaufbau und Rechenaufwand waren wie in dem vorhergehenden Beispiel zu dieser Kurve.

Abschliessend wollen wir das statistische Verhalten des größten Primfaktors der Ordnung der Jacobischen untersuchen.

**Beispiel 4.5.3** In [PPW03] schlagen Pelzl, Wollinger und Paar hyperelliptische Kurven von Geschlecht vier über Körpern mit 32 Bit für die Implementation von Kryptosystemen für ARM Prozessoren vor. Algorithmus 4.2.1 benötigt 40 Minuten und 2 MB auf einem Athlon 2300 MHz, um mit dem entsprechenden Hecke-Operator die Gruppenordnung der Jacobischen der Kurve zu bestimmen. In Anhang A sind je 100 Beispiele zu Hecke-Operatoren $T_p$ mit zufälligem $p \sim 2^{32}$ für die Kurven der Stufen $N = 23$, 47 und 53 aufgelistet, so dass man einen Überblick über die Verteilung der Divisorklassengruppen mit nahezu-Primzahlordnung bekommen kann.

Sei $N \in \{23, 47, 53\}$ und $n$ der größte Primteiler von $O := \#J_0(N)_p(\mathbb{F}_p)$ zu gegebener Primzahl $p$. Die folgende Tabelle listet den prozentualen Anteil der $p$ mit $n \sim O$, $n \sim \sqrt{O}$ und $n \sim \sqrt[3]{O}$ basierend auf Anhang A auf.

| $N$ | $\sim \sqrt[3]{O}$ | $\sim \sqrt{O}$ | $\sim O$ |
|---|---|---|---|
| 23 | 22% | 58% | 20% |
| 47 | 24% | 53% | 23% |
| 53 | 56% | 44% | 0% |

$X_0(53)$ lässt einen Morphismus von Grad zwei auf eine elliptische Kurve zu und hat daher keine fast-primen Gruppenordnungen $O$. Als Vergleich untersuchen wir nun das stochastische Zerfällungsverhalten einer zufälligen Zahl $m \in \mathbb{N}$. Nach [Ten95] hat $m$ durchschnittlich $\omega(m) := \log\log m$ Primteiler. Sei nun $q$ der größte Primteiler von $m$. Heuristisch gilt

$$\omega(m/q) = \omega(m) - 1$$

und daraus folgt

$$q = m^{1-1/e}$$

mit der Eulerschen Zahl $e$. Damit ist der größte Primteiler $q$ von $m$ etwa gleich $m^{0,632}$. Betrachten wir nur die Stufen $N = 23$ und $N = 47$, dann ist also das Zerfällungsverhalten von $O$ bezüglich des größten Primteilers ähnlich dem einer zufälligen Zahl.

# Kapitel 5

# Aufzählung koprimer Tupel

Motiviert durch den Algorithmus aus Kapitel 4 zur Berechnung von Hecke-Operatoren auf der Basis von Modulsymbolen untersuchen wir effiziente Implementationen zur Aufzählung koprimer Tupel. Ausgangspunkt ist die folgende Aufgabe:

**Problem 5.0.4** *Sei* $\tilde{x}, \tilde{y}, N, L \in \mathbb{N}$. *Bestimme alle Paare* $(x, y) \in \mathbb{N}^2$ *mit*

$$(\tilde{x}, \tilde{y}) \equiv (x, y) \bmod N \qquad \text{und} \qquad \mathrm{ggT}(x, y) = 1$$

*für* $\tilde{x} \leq x \leq L,\ \tilde{y} \leq y < x$. *Für jedes gefundene Paar* $(x, y)$ *mit* $\mathrm{ggT}(x, y) = 1$, *bestimme* $y^{-1} \bmod x$.

In unseren Anwendungen in Abschnitt 4.2 ist $L = \lfloor \sqrt{p} \rfloor$ für den Hecke-Operator $T_p$. Die Bezeichnungen von Problem 5.0.4 gelten für das gesamte Kapitel. Im folgenden untersuchen wir zwei Implementationen, die dieses Problem behandeln. Im ersten Abschnitt wird eine optimierte Version des Programms von Basmaji vorgestellt. Diese Methode rechnet den ggT direkt aus, wobei Primzahlen durch das Sieb des Erathostenes abgefangen werden.

Die zweite Methode benutzt eine Tabelle, in der zu jeder Zahl kleiner als $L$ die zugehörigen Primteiler abgespeichert sind. Diese Teiler werden ausgelesen und in einem Siebverfahren zur Aufzählung der koprimen Tupel verwendet. Wir stellen effiziente Speicherstrukturen und Siebverfahren vor.

Der letzte Abschnitt enthält experimentelle Ergebnisse beider Implementationen.

## 5.1 Aufzählung mittels ggT-Berechnung (Brute-Force)

Eine Möglichkeit zur Behandlung von Problem 5.0.4 ist das direkte Ausrechnen aller auftretenden ggT. Angefangen bei $\tilde{x}$ und $\tilde{y}$ wird durch zwei Schleifen iteriert, die die beiden Zahlen jeweils um den Wert $N$ erhöhen und darin den größten gemeinsamen Teiler des Tupels berechnen. Wir verwenden den maschinennahen binären ggT-Algorithmus zur Berechnung der ggT.

Für $L \in \mathbb{N}$, $10^4 \leq L \leq 10^{12}$ sind mindestens 7% aller Zahlen $\tilde{x} \leq L$ Primzahlen (s. Abschnitt 5.3). Falls das Inkrement $x$ von $\tilde{x}$ in der äusseren Schleife eine Primzahl ist, folgt für alle $\tilde{y} \leq y < x$ sofort $\mathrm{ggT}(x, y) = 1$. Daher benutzen wir das Sieb des Eratosthenes, um alle Primzahlen kleiner $L$ abzuspeichern. Ein Primzahltest kommt dann einem Tabellenaufruf gleich und wir reduzieren die Rechenzeit des Programms um etwa 7%. Algorithmus 5.1.1 beschreibt den Algorithmus in Pseudocode. Weiterführend wird nun das Erstellen des Siebes des Eratosthenes behandelt. Für Komplexitätsbetrachtungen, siehe Abschnitt 4.3.

---

**Algorithmus 5.1.1** Brute Force ggT-Aufzählung

**Eingabe** : $\tilde{x}, \tilde{y}, L, N \in \mathbb{N}$
**Ausgabe** : alle Tupel $(x, y)$ mit: $(x, y) \equiv (\tilde{x}, \tilde{y}) \bmod N$, $\mathrm{ggT}(x, y) = 1$, $\tilde{x} \leq x \leq L,\ \tilde{y} \leq y < x$.

```
1 begin
2     Primzahltabelle ← erstelle Sieb des Eratosthenes
3     for x ← x̃ to L step N do
4         if x ist als Primzahl in der Primzahltabelle aufgeführt then
5             for y ← ỹ to x step N do  return (x, y)
6         else
7             for y ← ỹ to x step N do
8                 if ggT(x, y) = 1 then  return (x, y)
9 end
```

---

### 5.1.1 Das Sieb des Eratosthenes

Das Sieb des Eratosthenes ist ein Verfahren zum Erstellen von Primzahltabellen. Dazu schreibt man alle natürlichen Zahlen von 2 bis zu einer Grenze $L \in \mathbb{N}$ in eine Liste und beginnt dann, alle Zahlen größer als 2 herauszustreichen, die durch 2 teilbar sind. Dann nimmt man iterativ die nächste ungestrichene Zahl und streicht dann alle Vielfachen davon, angefangen bei dem Quadrat der Zahl. Dies wird fortgeführt, bis die kleinste ungestrichene Zahl grösser ist als $\sqrt{L}$. Die verbleibenden Zahlen sind genau alle Primzahlen, die kleiner oder gleich $L$ sind.

Bei der Implementation dieses Verfahrens in `C++` bietet es sich an, einen Array des Character-Datentyps als Sieb zu verwenden. Der Index des Arrays repräsentiert dann die zu streichenden Zahlen, deren Character-Wert entweder 0 oder 1 annimmt, je nachdem ob die indizierte Zahl gestrichen ist.

Die Komplexität der Erstellung des Siebes ist $O(L)$ in Speicherplatz und Zeit. Da ein Character nur 8 bit hat und das Sieb als Vorberechnung nur einmal erstellt werden muss, fällt die Anwendung dieses Programmteils in der Komplexität nicht ins Gewicht. Für eine Verbesserung des Siebs des Eratosthenes siehe das „Sieb von Atkin“, [AtB99].

**Algorithmus 5.1.2** Der binäre ggT-Algorithmus

**Eingabe** : $u, v \in \mathbb{N}_0$
**Ausgabe** : $\mathrm{ggT}(u, v)$

```
1  begin
2      k ← 0
3      if u = 0 then
4          return v
5      if v = 0 then
6          return u
7      while u und v sind gerade do
8          verschiebe u um ein Bit nach rechts
9          verschiebe v um ein Bit nach rechts
10         k ← k + 1
       /* An diesem Punkt ist u oder v (oder beide) ungerade.  */
11     repeat
12         if u ist gerade then
13             verschiebe u um ein Bit nach rechts
14         else
15             if v ist gerade then
16                 verschiebe u um ein Bit nach rechts
17             else
                   /* u und v sind nun beide ungerade          */
18                 if u ≥ v then
19                     u ← (u − v)
20                     verschiebe u um ein Bit nach rechts
21                 else
22                     v ← (v − u)
23                     verschiebe v um ein Bit nach rechts
24     until u ≤ 0
25     verschiebe v um k Bit nach links
       /* übergebe v * 2^k                                     */
26     return v
27 end
```

### 5.1.2 Der binäre ggT-Algorithmus

Wir verwenden `gmp` für die Berechnung des ggT und des modularen Inversen. In der Auswertung in Abschnitt 5.3 werden wir feststellen, dass es zeitlich effizienter ist, diese beiden Rechenschritte getrennt durchzuführen, als sie mit Hilfe des erweiterten ggT-Algorithmus zusammen zufassen.

Der binäre ggT-Algorithmus ist ein Verfahren zur Berechnung des größten gemeinsamen Teilers zweier nichtnegativer ganzer Zahlen. Der Vorteil dieser

Methode gegenüber dem euklidischen Algorithmus liegt darin, dass Divisionen durch Bitverschiebungen ersetzt werden, die in der Binärdarstellung moderner Computersysteme effizienter durchgeführt werden können. Dieser Algorithmus wurde zuerst 1961 von Joseph Stein veröffentlicht.

Für $u, v \in \mathbb{N}_0$ berechnet der Algorithmus $\mathrm{ggT}(u, v)$ durch wiederholtes Anwenden der folgenden einfach nachzurechnenden Identitäten:

1. Falls $u = 0$, dann gilt $\mathrm{ggT}(u, v) = v$.
2. Falls $u$ gerade ist und $v$ ist gerade, dann gilt $\mathrm{ggT}(u, v) = 2\mathrm{ggT}(u/2, v/2)$.
3. Falls $u$ gerade ist und $v$ ungerade, dann gilt $\mathrm{ggT}(u, v) = \mathrm{ggT}(u/2, v)$.
4. Falls $u$ und $v$ beide ungerade sind, dann gilt $\mathrm{ggT}(u, v) = \mathrm{ggT}(|u - v|/2, v) = \mathrm{ggT}(u, |v - u|/2)$.

Im letzen Schritt wird jeweils die Version mit dem kleineren Operanden gewählt. Algorithmus 5.1.2 beschreibt eine Implementation des Algorithmus in Pseudocode. Im schlechtesten Fall benötigt der Algorithmus $O((\log_2 uv)^2)$ Bitperationen bei $\mathrm{ggT}(u, v) = 1$. Dies entspricht auch der Komplexität des euklidischen Algorithmus, in der Praxis ist der binäre ggT-Algorithmus jedoch etwa vierfach schneller.

## 5.2 Aufzählung mittels Sieben

Eine effiziente Alternative zur Behandlung von Problem 5.0.4 ist das Aussieben nicht-koprimer Zahlen $y$ in Kombination mit tabellarischem Abspeichern aller Teiler von $x$.

Man iteriert wieder durch eine Schleife $x \mapsto \tilde{x} + nN$, $n \in \mathbb{N}_0$ und sucht dann alle Primteiler $p_1, \ldots, p_{n_x}$ von $x$ aus einer vorerstellten Tabelle. Dann erstellt man eine Liste aller natürlichen Zahlen bis $x$ und streicht daraus alle Vielfachen von $p_1, \ldots, p_{n_x}$, ähnlich wie bei dem Sieb des Eratosthenes. Die verbleibenden Zahlen der Liste sind alle $y < x$ mit $\mathrm{ggT}(x, y) = 1$. Wie in Abschnitt 5.1 werden irreduzible Zahlen $x$ vorher durch das Sieb des Eratosthenes abgefangen, da alle $y < x$ bereits koprim sind.

Die folgenden beiden Abschnitte beinhalten effiziente Konzepte zur Erstellung und Zugriff auf die Teilertabelle, sowie zum Sieben der Teiler und wir zeigen, wie man ein Sieb des Erathostenes in die Speicherstruktur der Teilertabelle integrieren kann.

### 5.2.1 Die Teilertabelle

Der folgende Aufbau der Tabelle ermöglicht einen schnellen Zugriff auf die Primteiler bei minimalem Speicheraufwand. Wir benötigen einen *Teilerarray*, der die Primteiler speichert und einen *Steuerarray*, der zu gegebener Zahl $x$ die Grenzen des zugeordneten Bereichs im Teilerarray speichert. Ist $x$ eine Primzahl, wird kein Speicherplatz im Teilerarray benötigt, wenn wir im Steuerarray den Eintrag auf 0 setzen. So fungiert das Steuerarray gleichzeitig als Sieb des Eratosthenes (s. Abschnitt 5.1.1).

- Das Steuerarray ist ein `integer`-Array $S$ der Länge $L$.
- Ist $x$ eine Primzahl, so ist $S[x] = 0$.
- Ist $x$ keine Primzahl, so ist $S[x]$ die Addresse des letzten zu $x$ gehörigen Eintrags im Teilerarray.

Das Teilerarray speichert die Primteiler in einem ähnlichen Format wie das Sieb des Eratosthenes aus Abschnitt 5.1.1. Die Einträge beziehen sich hier allerdings ausschliesslich auf die Primzahlen, d.h. der erste Eintrag bezieht sich auf die 2, der zweite auf die 3, der dritte auf die 5 usw. Will man nun wissen, zu welcher Primzahl der $n$-te zu $x$ gehörige Eintrag im Teilerarray gehört, so muss man mit dem Sieb des Eratosthenes (der im Steuerarray enthalten ist) $n$ Primzahlen vorwärtszählen und erhält dann die dem Eintrag entsprechende Primzahl. Um weiteren Speicherplatz zu sparen, repräsentiert jeweils ein Bit einen Eintrag im Teilerarray. Beispiel 5.2.1 illustriert das Vorgehen.

- Das Teilerarray $T$ ist ein `char`-Array, deren Länge im nachfolgenden errechnet wird.
- Die einzelnen Bits der `char`-Einträge werden gesetzt, wenn die entsprechende Primzahl ein Faktor von $x$ ist.

Die Länge des Teilerarrays setzt sich zusammen aus allen auftretenden Primteilern aller Zahlen bis $L$. Hinzu kommt, dass die Einträge immer auf ein Vielfaches von 8 aufgefüllt werden müssen, da der `char`-Datentyp als Grundlage verwendet wird. Teilt man das gesamte Ergebnis durch 8 hat man die Anzahl der nötigen `char`-Blöcke und damit die Länge des Teilerarrays.

**Beispiel 5.2.1** *Wir erläutern das Konzept mit $L = 10$. Das Steuerarray $S$ markiert entweder das jeweils letzte Byte im Teilerarray $T$, oder der Eintrag ist 0, falls der Index eine Primzahl ist. Die Einträge von $T$ repräsentieren 8 bit. Damit wir nicht während des Auslesens Zeit mit Indexverschiebungen verlieren, lassen wir die ersten Einträge von $S$ und $T$ unbelegt.*

| | | | |
|---|---|---|---|
| $S[0]$ : | *unbelegt* | $T[0]$ : | *unbelegt* |
| $S[1]$ : | *unbelegt* | $T[1]$ : | 10000000 |
| $S[2]$ : | 0 | $T[2]$ : | 11000000 |
| $S[3]$ : | 0 | $T[3]$ : | 10000000 |
| $S[4]$ : | 1 | $T[4]$ : | 01000000 |
| $S[5]$ : | 0 | $T[5]$ : | 10100000 |
| $S[6]$ : | 2 | | |
| $S[7]$ : | 0 | | |
| $S[8]$ : | 3 | | |
| $S[9]$ : | 4 | | |
| $S[10]$ : | 5 | | |

*Z.B. zeigt uns $S[7] = 0$ an, dass 7 eine Primzahl ist. Wollen wir die Teiler von 10 ermitteln, so ermitteln wir aus $S[10] = 5$ und $S[9] = 4$, dass 5 die Start-*

*und Endaddresse der Primteilerblöcke im Teilerarray $T$ ist. Die Bitbelegung von $T[5]$ ist* 10100000*, d.h. die erste und die dritte Primzahl treten in* 10 *auf. Nun iterieren wir wieder durch den Steuerarray und Zählen die Nullen. Die erste Null finden wir bei $S[2]$, d.h.* 2 *ist die erste Primzahl und damit ein Teiler von* 10*, die dritte Null steht bei $S[5]$ und damit ist* 5 *der andere Teiler.*

*Hier sieht man auch, wie die Grösse von $T$ ermittelt wird. Es treten für $L \leq 10$ nur* 5 *Zahlen auf, die keine Primteiler sind. Der maximal auftretende Primteiler ist der dritte, weswegen alle Einträge nur auf* 8 *bit aufgefüllt werden müssen. Also werden insgesamt* 40 *bit benötigt. Das ergibt eine Länge von* 5 *für das Teilerarray $T$, plus einem unbelegten Byte aus Indizierungsgründen.*

Das **Erstellen** der Teilertabelle durchläuft folgende Schritte: zuerst wird das Sieb des Eratosthenes erstellt. Dann wird eine Schleife von 2 bis $L$ durchlaufen, in der jede Zahl $a$ mit Hilfe des Siebes des Eratosthenes auf Irreduziblität getestet wird. Ist die Zahl eine Primzahl, erhält das Steuerarray den Eintrag 0. Ist Sie keine Primzahl, so wird durch Ausprobieren der kleinste Primteiler $p$ gesucht, abgespalten und in die Teilertabelle eingetragen. Per Induktion folgt, dass die sich ergebende Zahl $a/p$ bereits in der Teilertabelle abgespeichert ist. Nun muss man nur noch die Teiler von $a/p$ aus der Liste auslesen und in den Abschnitt von $a$ in der Teilertabelle übertragen. Dies geht sehr effizient, indem man logische ODER-Operationen auf den betreffenden `char`-Blöcken ausführt. Abschliessend trägt man das Ende des Speicherbereichs in den Steuerarray ein.

Der Speicheraufwand kann um etwa die Hälfte reduziert werden, wenn man die geraden Zahlen nicht in die Teilertabelle einträgt. Ist $a$ gerade, so kann man eine schnelle *Bitverschiebung* nach rechts durchführen, bis $a'$ keine Teiler von 2 mehr enthält und dann die Teiler von $a'$ auslesen. Die auftretenden Bitverschiebungen fallen im Gesamtaufwand nur marginal ins Gewicht.

Zum **Auslesen** der Teiler einer Zahl $a$ muss man zuerst ermitteln, wo der entsprechende Addressbereich für $a$ im Teilerarray beginnt. Dies ist entweder $S[a-1]+1$ oder $S[a-2]+1$, wenn $a-1$ eine Primzahl ist. Hat man den Anfangsbereich von $a$ im Teilerarray gefunden, so iteriert man durch die `char`-Blöcke und überprüft, welche Bits belegt sind, während man gleichzeitig mit Hilfe des Siebs des Eratosthenes (Steuerarray) die Primzahlen hochzählt. Ist das $n$-te bit belegt, hat man die $n$-te Primzahl als Teiler von $a$ identifiziert. Unbelegte `char`-Blöcke sind gleich null und können daher mit einem Test schnell übersprungen werden.

Beim Hochzählen der Primzahlen bietet es sich an, nach den ersten drei Primzahlen den Zähler abwechselnd um +2 und +4 zu inkrementieren. Dadurch überspringt man alle Vielfachen von 2 und 3.

### 5.2.2 Aussieben der Primteiler

Das Sieben erfolgt wie beim Sieb des Erathostenes. Man erstellt ein `char`-Array der Länge $L$ und setzt die Einträge von $\tilde{y}$ bis $x$ auf 1. In `C++` benutzen wir dafür die Methode `memset`. Dann streichen wir aus diesem Intervall alle Vielfachen der ausgelesenen Primzahlen wie beim Sieb des Erathostenes. Es lohnt sich, für die gegebenen Untergrenzen $\tilde{x}$ und $\tilde{y}$ eine Vorabberechnung zu machen:

**Lemma 5.2.2** *Sei $\tilde{x}, \tilde{y}, N \in \mathbb{N}$ gegeben. Gibt es einen Primteiler $p$ von $ggT(\tilde{x}, N)$, dann gilt $p|\tilde{x} + aN$ für alle $a \in \mathbb{Z}$ und*

$$p|\tilde{y} \iff p|\mathrm{ggT}(\tilde{x} + aN, \tilde{y} + bN) \text{ für alle } a, b \in \mathbb{Z}.$$

Der Beweis ist trivial. Sammeln wir also vor Beginn des Algorithmus die gemeinsamen Teiler von $\tilde{x}$ und $N$ aus der Teilertabelle, so gibt es keine teilerfremden Tupel, falls einer dieser Primteiler auch in $\tilde{y}$ enthalten ist und wir brechen den Algorithmus ab. Falls keiner dieser Primteiler in $\tilde{y}$ enhalten ist, wissen wir, dass diese zwar für jedes $x$ auftreten, aber niemals ausgesiebt werden müssen. Da diese Teiler im Verhältnis zu $L$ ziemlich klein sind, sparen wir dadurch viele Sieboperationen. Das Lemma ist nicht anwendbar, falls man mehrere Stufen $N_i$ simultan ausrechnen will, wie in Abschnitt 4.2.3 beschrieben. Für eine Diskussion der Komplexität, siehe Abschnitt 4.3.

## 5.3 Experimentelle Ergebnisse

Ist Speicher kein kritischer Faktor, empfiehlt es sich generell, die Siebmethode zu verwenden. Das Langzahlpaket `gmp` bietet die Möglichkeit, für zwei gegebene ganze Zahlen $a$, $b$ das Inverse $a^{-1}$ mod $b$ zu berechnen bzw. anzuzeigen, dass $\mathrm{ggT}(a, b) \neq 1$. Der geringe Anteil teilerfremder Tupel bedingt jedoch, dass man bei der Brute-Force Methode besser die modulare Inversion getrennt von der Bestimmung teilerfremder Tupel berechnet, da das Zeitverhältnis von modularer Inversion zu binärem ggT einer reinen Verwendung der modularen Inversion ineffizient macht:

Bei unseren Messungen war der binäre ggT Algorithmus konstant vier mal schneller als die modulare Inversion für zufällige Eingabeparameter bis $L \leq 10^{13}$. Der Anteil teilerfremder Tupel liegt nach Lemma 4.3.1 approximativ bei 61%. Bezeichnen wir mit $t$ die benötigte Zeit für die Durchführung einer modularen Inversion, so benötigen wir durchschnittlich $0,25t + 0,61t = 0,86t$, um die teilerfremden Tupel mit dem binären ggT Algorithmus zu identifizieren und dann separat die modulare Inversion durchzuführen, wir sparen also 14% der Zeit.

So ist der Anteil zu berechnender Inversionen bei der Brute-Force und der Siebmethode gleich und es genügt, die Zeiten für die Erstellung der Liste teilerfremder Tupel zu betrachten. Die Zeiten wurden auf einem 1GHz Pentium III mit 2GB RAM unter Linux in `C++` mit `gcc` 4.1.1 erstellt. Modulare Inversion und binärer ggT wurden dem Langzahlpaket `gmp` entnommen. Zur Zeitmessung der zwei Algorithmen wurde Problem 5.0.4 zu vorgegebener Proportion von $L$ jeweils $10^4$ mal mit zufälligen Initialbedingungen $\tilde{x}$, $\tilde{y}$ und $1 \leq N \leq 500$ bearbeitet und der Mittelwert der resultierenden Zeiten gebildet. Mit Hinblick auf Abschnitt 4.4 wurde $L \leq 3000000$ getestet, da $3000000^2$ eine 43-Bit Zahl ist. Abbildung 5.1 zeigt die interpolierten Werte, dabei ist die Brute-Force Methode mit **BF** und das Siebverfahren mit **SV** bezeichnet. Mit dem Verwenden des Siebverfahrens gewinnt man zeitlich einen Faktor von etwa 2,5 gegenüber Brute-Force.

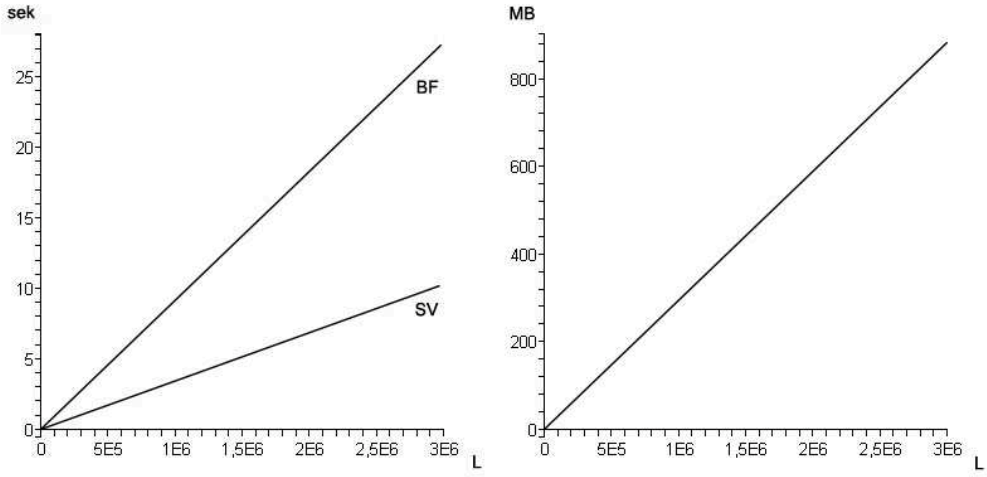


Abbildung 5.1: Zeitbedarf in Sekunden und Speicherbedarf der Teilertabelle in MB in Abhängigkeit von $L$ für Brute-Force(BF) und das Siebverfahren(SV).

Wie man bereits an dem Anteil der Primzahlen erahnen kann, spart man etwa 7% der Zeit, wenn man vor der ggT-Berechnung oder vor dem Sieben Primzahlen mit dem Sieb des Erathostenes abfängt, wie in Abschnitt 5.1 beschrieben. Für die Brute-Force Methode sind die koprimen Tupel die Aufwändigsten, daher ist eine Verwendung dieses Tests sehr zu empfehlen. Der Speicheraufwand des Siebs des Erathostenes ist mit etwa $L$ Byte durchaus auch für grössere Zahlen zu verwenden.

| $L \sim$ | Primzahlen |
|---|---|
| $1 \cdot 10^5$ | $9,6\%$ |
| $2 \cdot 10^5$ | $9\%$ |
| $3 \cdot 10^5$ | $8,7\%$ |
| $4 \cdot 10^5$ | $8,5\%$ |
| $5 \cdot 10^5$ | $8,3\%$ |
| $1 \cdot 10^6$ | $7,8\%$ |
| $2 \cdot 10^6$ | $7,5\%$ |
| $3 \cdot 10^6$ | $7,2\%$ |

Für $L$ im Bereich von 30 Bit würde diese Methode etwa 1GB RAM belegen. Würde man eine Kompressionsmethode, wie die aus Abschnitt 5.2.1 verwenden, könnte man noch Zahlen $L$ im Bereich von 34 Bit mit einer RAM Anforderung von 2GB beschleunigen. Man kann leicht nachrechnen, dass der Speicheraufwand durch Weglassen der ersten sieben Primzahlen aus der Teilertabelle um etwa 90% reduziert wird. Damit könnte man Zahlen bis zu 39 Bit mit 6,4GB Speicher bearbeiten, hätte jedoch jedesmal auf Teilbarkeit durch 2,3,5,7,11,13 und 17 zu testen, was nicht mehr den vollen Geschwindigkeitsvorteil einbringen würde.

### 5.3.1 Kleines $L$

Das Siebverfahren zeichnet sich vor allem für kleines $L$ durch hohe Geschwindigkeit aus. Daher wurde eine zweite Versuchsreihe mit Schwerpunkt auf diese Grössenordnung erstellt. Da der Speicheraufwand für die Teilertabelle vernachlässigbar ist, wurde auf eine Kompression wie in 5.2.1 verzichtet. Die Teilertabelle wurde als Matrix gespeichert, deren $(i, j)$-tes Byte mit 0,1 oder 2 belegt

ist. $i$ ist dabei die $i$-te Primzahl[1] und $j$ die zu zerlegende Zahl. Die Belegung 1 bzw. 0 indiziert, ob die $i$-te Primzahl $j$ teilt, oder nicht. Eine 2 indiziert, dass es die grösste in $j$ enthaltene Primzahl ist und das Auslesen gestoppt werden kann. $N$ variiert in dieser Versuchsreihe zwischen 1 und 100, ansonsten sind die Algorithmen und deren Ausführung identisch mit den vorher beschriebenen.

| $L \sim$ | Brute-Force | Sieben | Speicher | Primzahlen |
|---|---|---|---|---|
| 500 | 5.1e-05 sec | 7e-06 sec | $< 1$ MB | 19% |
| 1000 | 2.1e-04 sec | 1.7e-05 sec | $< 1$ MB | 17% |
| 5000 | 6.1e-03 sec | 2.8e-04 sec | $< 1$ MB | 13% |
| 10000 | 2.6e-02 sec | 1.2e-03 sec | $< 1$ MB | 12% |
| 50000 | 7.3e-01 sec | 3.1e-02 sec | $< 1$ MB | 10% |

Das Aufzählen der teilerfremden Tupel durch Sieben ist in diesem Grössenbereich deutlich effizienter, wie man an der Tabelle erkennen kann.

[1] Bsp: 2 ist die erste Primzahl, 3 die zweite usw.

# Anhang A

# 32-Bit Beispiele zum charakteristischen Polynom des Heckeoperators

Wir führen nun jeweils 100 Beispiele zu charakteristischen Polynomen $\chi_{T_p}$ von Hecke-Operatoren $T_p$ mit $p \sim 2^{32}$ prim zu den Modulkurven der Stufen $N = 23, 47$ und 53 auf und werten diese in $\chi_{T_p}(p+1) = \#J_0(N)_p(\mathbb{F}_p)$ aus, um die statistische Verteilung großer Primteiler von $\#J_0(N)_p(\mathbb{F}_p)$ zu veranschaulichen. Die zugrundeliegenden Algorithmen wurden in `C++` programmiert, wobei die Modulsymbolreduktion mit `Magma` erfolgte.

## A.1 Beispiele zur Stufe $N = 23$

Die Modulkurve $X_0(23)$ ist eine hyperelliptische Kurve von Geschlecht $g = 2$. Diese ist gegeben, durch die affine Gleichung

$$\begin{aligned} y^2 &= x^6 - 8x^5 + 2x^4 + 2x^3 - 11x^2 + 10x - 7 \\ &= (x^3 - x + 1)(x^3 - 8x^2 + 3x - 7) \end{aligned}$$

Entnommen ist diese Kurve aus [Gal96], Kap. 4, Tabelle 3.

| $p = 4294970839$ | $\chi_{T_p}(x) = x^2 - 81420x - 582482025$ |
|---|---|
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 5^2 \cdot 11 \cdot 151 \cdot 16013441 \cdot 27740731$ |
| $p = 4294971391$ | $\chi_{T_p}(x) = x^2 + 77996x - 6806436496$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^4 \cdot 5^2 \cdot 11^2 \cdot 381138724005799$ |
| $p = 4294971943$ | $\chi_{T_p}(x) = x^2 + 17600x - 1849637120$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^6 \cdot 3^2 \cdot 11 \cdot 29574331 \cdot 98444701$ |
| $p = 4294975117$ | $\chi_{T_p}(x) = x^2 + 136034x + 4514731409$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 5 \cdot 11 \cdot 335407191479982679$ |
| $p = 4294975393$ | $\chi_{T_p}(x) = x^2 - 100514x - 1877558831$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 11 \cdot 142361071 \cdot 11779511069$ |

| **Beispiele zur Stufe $N = 23$ (Fortsetzung)** | |
|---|---|
| $p = 4294976083$ | $\chi_{T_p}(x) = x^2 - 151052x + 3988113296$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 11 \cdot 104807788644276409$ | |
| $p = 4294976221$ | $\chi_{T_p}(x) = x^2 + 44168x - 537322564$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 11 \cdot 379 \cdot 1061 \cdot 260650601839$ | |
| $p = 4294976359$ | $\chi_{T_p}(x) = x^2 + 168052x + 6488453551$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 11 \cdot 1677049428983141261$ | |
| $p = 4294976773$ | $\chi_{T_p}(x) = x^2 - 78574x - 3801673711$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 11 \cdot 19 \cdot 499 \cdot 176875166715179$ | |
| $p = 4294979809$ | $\chi_{T_p}(x) = x^2 + 34930x - 2894917795$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 5 \cdot 11 \cdot 269 \cdot 271 \cdot 4600886553889$ | |
| $p = 4294979947$ | $\chi_{T_p}(x) = x^2 + 175076x + 7192645424$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 11 \cdot 131 \cdot 241 \cdot 3320006836831$ | |
| $p = 4294981189$ | $\chi_{T_p}(x) = x^2 - 74820x + 980346420$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^6 \cdot 5 \cdot 11 \cdot 1471 \cdot 178831 \cdot 19921261$ | |
| $p = 4294982431$ | $\chi_{T_p}(x) = x^2 - 85504x + 820243379$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 5^2 \cdot 11 \cdot 2081 \cdot 32233640913601$ | |
| $p = 4294982569$ | $\chi_{T_p}(x) = x^2 + 94358x + 1167345541$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 3^2 \cdot 11 \cdot 211 \cdot 1409 \cdot 108359 \cdot 5784139$ | |
| $p = 4294982983$ | $\chi_{T_p}(x) = x^2 - 84156x + 845077959$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 11 \cdot 349 \cdot 4805031879423449$ | |
| $p = 4294983811$ | $\chi_{T_p}(x) = x^2 + 45500x - 465940625$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 3^2 \cdot 11^2 \cdot 13109 \cdot 1040419 \cdot 1242001$ | |
| $p = 4294984501$ | $\chi_{T_p}(x) = x^2 - 24078x - 3797494479$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 11 \cdot 109 \cdot 15385144665542831$ | |
| $p = 4294985467$ | $\chi_{T_p}(x) = x^2 + 26084x - 381681361$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 5^2 \cdot 11^2 \cdot 14401 \cdot 423455721919$ | |
| $p = 4294986019$ | $\chi_{T_p}(x) = x^2 - 127448x + 3878158931$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 11 \cdot 29 \cdot 2731 \cdot 21173772314039$ | |
| $p = 4294986433$ | $\chi_{T_p}(x) = x^2 - 72368x - 2186614964$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 5 \cdot 11^2 \cdot 19 \cdot 100296855407129$ | |
| $p = 4294970377$ | $\chi_{T_p}(x) = x^2 - 62142x - 353118564$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 3487699 \cdot 120204960149$ | |
| $p = 4294972861$ | $\chi_{T_p}(x) = x^2 - 39474x - 1954279836$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 5 \cdot 11 \cdot 829 \cdot 11131 \cdot 9086684779$ | |
| $p = 4294974517$ | $\chi_{T_p}(x) = x^2 - 41114x - 1330537876$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 7682639 \cdot 54569996081$ | |
| $p = 4294974793$ | $\chi_{T_p}(x) = x^2 - 68x - 589045424$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 5 \cdot 11 \cdot 61 \cdot 3259 \cdot 75109 \cdot 5615551$ | |
| $p = 4294976311$ | $\chi_{T_p}(x) = x^2 + 24706x - 1448916436$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 5 \cdot 11^2 \cdot 271 \cdot 499 \cdot 91921 \cdot 613231$ | |
| $p = 4294978381$ | $\chi_{T_p}(x) = x^2 - 90370x - 609529420$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11^2 \cdot 31 \cdot 13841 \cdot 226451 \cdot 392251$ | |

| **Beispiele zur Stufe $N = 23$ (Fortsetzung)** | |
|---|---|
| $p = 4294981003$ | $\chi_{T_p}(x) = x^2 + 119898x + 2750571756$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 131 \cdot 3200447048421901$ | |
| $p = 4294981417$ | $\chi_{T_p}(x) = x^2 + 124978x + 3049573916$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11^3 \cdot 1021 \cdot 1541581 \cdot 2201431$ | |
| $p = 4294981969$ | $\chi_{T_p}(x) = x^2 + 116394x + 1942740684$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11^2 \cdot 191 \cdot 185161 \cdot 1077720671$ | |
| $p = 4294983073$ | $\chi_{T_p}(x) = x^2 + 50926x - 4406881676$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 419252234723809871$ | |
| $p = 4294985143$ | $\chi_{T_p}(x) = x^2 - 87884x + 1923383984$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 11 \cdot 104809772321759599$ | |
| $p = 4294986247$ | $\chi_{T_p}(x) = x^2 + 115866x + 2832661044$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 139 \cdot 3016253191010201$ | |
| $p = 4294987903$ | $\chi_{T_p}(x) = x^2 - 56338x + 622906156$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 3^2 \cdot 5 \cdot 11^2 \cdot 181 \cdot 269 \cdot 59141 \cdot 294131$ | |
| $p = 4294988179$ | $\chi_{T_p}(x) = x^2 - 84642x + 1377653796$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 71 \cdot 5904788710412239$ | |
| $p = 4294993837$ | $\chi_{T_p}(x) = x^2 - 45286x + 386050004$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 5 \cdot 11 \cdot 131 \cdot 640068617826269$ | |
| $p = 4294995769$ | $\chi_{T_p}(x) = x^2 + 36206x + 191737484$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 19 \cdot 149419 \cdot 147678420581$ | |
| $p = 4294996183$ | $\chi_{T_p}(x) = x^2 + 137634x + 4734494244$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 29 \cdot 1051 \cdot 1055881 \cdot 13027801$ | |
| $p = 4294996597$ | $\chi_{T_p}(x) = x^2 + 96736x + 2213561804$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 11^2 \cdot 9528621518510351$ | |
| $p = 4294997149$ | $\chi_{T_p}(x) = x^2 + 75892x + 82276496$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11^2 \cdot 61 \cdot 251 \cdot 1249 \cdot 5101 \cdot 390721$ | |
| $p = 4294998667$ | $\chi_{T_p}(x) = x^2 + 8160x - 6491464020$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 419251104518623661$ | |
| $p = 4294971349$ | $\chi_{T_p}(x) = x^2 - 17708x - 1881806684$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^6 \cdot 11^2 \cdot 3331 \cdot 715119753769$ | |
| $p = 4294972039$ | $\chi_{T_p}(x) = x^2 + 105248x + 2680580656$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 3^2 \cdot 11 \cdot 2998201 \cdot 3884323589$ | |
| $p = 4294972867$ | $\chi_{T_p}(x) = x^2 + 144328x + 4652606416$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^8 \cdot 11 \cdot 6550927495153709$ | |
| $p = 4294973281$ | $\chi_{T_p}(x) = x^2 - 4396x - 4263626416$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 211 \cdot 45319 \cdot 43843482041$ | |
| $p = 4294975627$ | $\chi_{T_p}(x) = x^2 - 88938x + 888646716$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 859 \cdot 4759 \cdot 12379 \cdot 8284481$ | |
| $p = 4294976731$ | $\chi_{T_p}(x) = x^2 - 78546x + 1133308404$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11^3 \cdot 31 \cdot 3391 \cdot 32959920689$ | |
| $p = 4294977559$ | $\chi_{T_p}(x) = x^2 + 23300x - 1776072080$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 3^2 \cdot 5 \cdot 11 \cdot 2329158120215231$ | |

| **Beispiele zur Stufe $N = 23$ (Fortsetzung)** | |
|---|---|
| $p = 4294978249$ | $\chi_{T_p}(x) = x^2 - 4702x - 342667924$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11^2 \cdot 19 \cdot 449 \cdot 4099 \cdot 9091 \cdot 119891$ | |
| $p = 4294979077$ | $\chi_{T_p}(x) = x^2 - 101012x - 420090844$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 11^2 \cdot 229 \cdot 349 \cdot 409 \cdot 291489101$ | |
| $p = 4294980733$ | $\chi_{T_p}(x) = x^2 - 8748x - 1185530544$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 5 \cdot 11 \cdot 83849190598882669$ | |
| $p = 4294984321$ | $\chi_{T_p}(x) = x^2 + 43142x - 1810603084$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11^2 \cdot 3259 \cdot 791311 \cdot 14779189$ | |
| $p = 4294985287$ | $\chi_{T_p}(x) = x^2 - 24730x - 1360732780$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 562271219 \cdot 745628209$ | |
| $p = 4294987357$ | $\chi_{T_p}(x) = x^2 - 40446x - 826150716$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 5 \cdot 11^2 \cdot 139 \cdot 54838999612031$ | |
| $p = 4294987771$ | $\chi_{T_p}(x) = x^2 + 59134x - 4921783556$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 211 \cdot 9419 \cdot 210955044031$ | |
| $p = 4294989151$ | $\chi_{T_p}(x) = x^2 - 51836x - 7321058896$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 11 \cdot 1579 \cdot 1994459 \cdot 33281201$ | |
| $p = 4294989289$ | $\chi_{T_p}(x) = x^2 + 44020x - 4078176220$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 5 \cdot 11 \cdot 71 \cdot 172169 \cdot 1714875689$ | |
| $p = 4294989703$ | $\chi_{T_p}(x) = x^2 - 108732x - 313973964$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 28032761 \cdot 14955285511$ | |
| $p = 4294991359$ | $\chi_{T_p}(x) = x^2 - 11184x - 484045056$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^8 \cdot 11 \cdot 510101 \cdot 12842058169$ | |
| $p = 4294991497$ | $\chi_{T_p}(x) = x^2 - 113244x + 2963425104$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 2341 \cdot 77801 \cdot 2301833449$ | |
| $p = 4294992463$ | $\chi_{T_p}(x) = x^2 - 177292x + 7648290896$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 11 \cdot 71 \cdot 35899 \cdot 100049 \cdot 410999$ | |
| $p = 4294969501$ | $\chi_{T_p}(x) = x^2 + 154536x + 5547327804$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^4 \cdot 5 \cdot 7^2 \cdot 11 \cdot 4315379 \cdot 99137531$ | |
| $p = 4294970467$ | $\chi_{T_p}(x) = x^2 - 37204x - 2363059241$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 3^2 \cdot 11 \cdot 361279 \cdot 515749349891$ | |
| $p = 4294972951$ | $\chi_{T_p}(x) = x^2 + 58300x + 351620695$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 11 \cdot 24851 \cdot 716981 \cdot 94120139$ | |
| $p = 4294974331$ | $\chi_{T_p}(x) = x^2 + 39264x - 6837244821$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 11 \cdot 71 \cdot 89 \cdot 265389706995839$ | |
| $p = 4294975297$ | $\chi_{T_p}(x) = x^2 - 151346x + 5264642429$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 5^3 \cdot 11 \cdot 71 \cdot 188949170687461$ | |
| $p = 4294975849$ | $\chi_{T_p}(x) = x^2 - 103924x + 1830960164$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^8 \cdot 11 \cdot 761 \cdot 1709 \cdot 2111 \cdot 2385961$ | |
| $p = 4294975987$ | $\chi_{T_p}(x) = x^2 + 119608x + 516922571$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 11 \cdot 6709 \cdot 65921 \cdot 3791921261$ | |
| $p = 4294976263$ | $\chi_{T_p}(x) = x^2 - 72308x - 2959885129$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 5 \cdot 11 \cdot 59 \cdot 97879 \cdot 137339 \cdot 422879$ | |

| **Beispiele zur Stufe $N = 23$ (Fortsetzung)** | |
|---|---|
| $p = 4294976677$ | $\chi_{T_p}(x) = x^2 - 25666x - 91642111$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 5^2 \cdot 11 \cdot 31 \cdot 14389 \cdot 150381365609$ |
| $p = 4294977091$ | $\chi_{T_p}(x) = x^2 - 52928x - 2100764509$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 11 \cdot 59 \cdot 3851 \cdot 7380709908721$ |
| $p = 4294978057$ | $\chi_{T_p}(x) = x^2 - 38006x - 4952034011$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 5 \cdot 11 \cdot 31 \cdot 3691 \cdot 2931228180271$ |
| $p = 4294978333$ | $\chi_{T_p}(x) = x^2 - 202658x + 10084069421$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 5 \cdot 11 \cdot 31 \cdot 191 \cdot 56642669352211$ |
| $p = 4294979023$ | $\chi_{T_p}(x) = x^2 + 75432x - 226249749$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 5 \cdot 11 \cdot 19 \cdot 479 \cdot 36853430282849$ |
| $p = 4294979851$ | $\chi_{T_p}(x) = x^2 + 18448x - 195568144$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^6 \cdot 11 \cdot 349 \cdot 75080307219811$ |
| $p = 4294981231$ | $\chi_{T_p}(x) = x^2 + 76604x + 913901584$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^4 \cdot 3^2 \cdot 11 \cdot 7951 \cdot 1464715765129$ |
| $p = 4294981783$ | $\chi_{T_p}(x) = x^2 - 139052x + 1885058551$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 11 \cdot 19 \cdot 141991 \cdot 219001 \cdot 2838281$ |
| $p = 4294982473$ | $\chi_{T_p}(x) = x^2 + 90774x - 1124298711$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 11^2 \cdot 152456729946958321$ |
| $p = 4294983163$ | $\chi_{T_p}(x) = x^2 + 30092x - 250491664$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^4 \cdot 5 \cdot 11 \cdot 12453659 \cdot 1683241121$ |
| $p = 4294984957$ | $\chi_{T_p}(x) = x^2 - 40650x - 2055091995$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 11 \cdot 1307461 \cdot 1282619255339$ |
| $p = 4294985647$ | $\chi_{T_p}(x) = x^2 + 26428x - 4646834249$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 11^2 \cdot 19 \cdot 541 \cdot 14831663706361$ |
| $p = 4294971301$ | $\chi_{T_p}(x) = x^2 + 208356x + 10581618564$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^8 \cdot 5 \cdot 11 \cdot 31 \cdot 89 \cdot 541 \cdot 877788689$ |
| $p = 4294971991$ | $\chi_{T_p}(x) = x^2 - 62750x - 4984467380$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^2 \cdot 11^2 \cdot 41 \cdot 929576441120111$ |
| $p = 4294972267$ | $\chi_{T_p}(x) = x^2 - 81766x - 187365116$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^2 \cdot 3^2 \cdot 5 \cdot 11 \cdot 8501 \cdot 195599 \cdot 5602871$ |
| $p = 4294973233$ | $\chi_{T_p}(x) = x^2 + 25388x - 422926684$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^8 \cdot 11^2 \cdot 15161 \cdot 39279893999$ |
| $p = 4294973923$ | $\chi_{T_p}(x) = x^2 - 82168x - 324925424$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^6 \cdot 3^2 \cdot 5 \cdot 11^3 \cdot 31 \cdot 186619 \cdot 831811$ |
| $p = 4294975717$ | $\chi_{T_p}(x) = x^2 - 126856x + 3715155664$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^2 \cdot 3^2 \cdot 5 \cdot 11 \cdot 497051 \cdot 18743144621$ |
| $p = 4294976131$ | $\chi_{T_p}(x) = x^2 + 96550x - 366063020$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^2 \cdot 11 \cdot 1559 \cdot 60509 \cdot 4444393511$ |
| $p = 4294976269$ | $\chi_{T_p}(x) = x^2 - 193726x + 8698341724$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^2 \cdot 3^2 \cdot 7^2 \cdot 11^3 \cdot 7856431203581$ |
| $p = 4294977097$ | $\chi_{T_p}(x) = x^2 + 134256x + 2941841664$ |
| $\#J_0(23)_p(\mathbb{F}_p)$ | $= 2^2 \cdot 11^3 \cdot 12809 \cdot 270509179891$ |

| **Beispiele zur Stufe $N = 23$ (Fortsetzung)** | |
|---|---|
| $p = 4294977787$ | $\chi_{T_p}(x) = x^2 - 167086x + 6742416724$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 3^2 \cdot 5^2 \cdot 11^3 \cdot 29 \cdot 211 \cdot 2516540539$ | |
| $p = 4294978201$ | $\chi_{T_p}(x) = x^2 - 25294x - 447145196$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 3^2 \cdot 5 \cdot 11 \cdot 1009 \cdot 9233428996621$ | |
| $p = 4294979167$ | $\chi_{T_p}(x) = x^2 + 43462x - 10535664844$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 7^2 \cdot 11 \cdot 149 \cdot 181 \cdot 136711 \cdot 2320649$ | |
| $p = 4294979443$ | $\chi_{T_p}(x) = x^2 + 15458x - 185052604$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 29 \cdot 101 \cdot 109 \cdot 306419 \cdot 4285579$ | |
| $p = 4294979719$ | $\chi_{T_p}(x) = x^2 - 89684x + 1071806884$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 31 \cdot 13523801763432461$ | |
| $p = 4294980961$ | $\chi_{T_p}(x) = x^2 + 63124x + 995286224$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11^2 \cdot 31 \cdot 9689 \cdot 126894517201$ | |
| $p = 4294981927$ | $\chi_{T_p}(x) = x^2 + 36126x - 1913514156$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 269 \cdot 571 \cdot 242329 \cdot 11263669$ | |
| $p = 4294982617$ | $\chi_{T_p}(x) = x^2 + 40594x - 1899496796$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 5 \cdot 11 \cdot 1061 \cdot 79029432086141$ | |
| $p = 4294982893$ | $\chi_{T_p}(x) = x^2 + 148042x + 4388491636$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 5 \cdot 11 \cdot 199 \cdot 569 \cdot 740542218281$ | |
| $p = 4294983997$ | $\chi_{T_p}(x) = x^2 + 11974x - 5369036$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 5 \cdot 11^2 \cdot 79 \cdot 191 \cdot 505182719479$ | |
| $p = 4294984411$ | $\chi_{T_p}(x) = x^2 - 11776x - 10917467476$ |
| $\#J_0(23)_p(\mathbb{F}_p) = 2^2 \cdot 11 \cdot 2161 \cdot 471391 \cdot 411560099$ | |

## A.2 Beispiele zur Stufe $N = 47$

Die Modulkurve $X_0(47)$ ist eine hyperelliptische Kurve von Geschlecht $g = 4$ mit reeller Multiplikation. Diese ist gegeben, durch die affine Gleichung

$$y^2 = x^{10} + 6x^9 + 11x^8 + 24x^7 + 19x^6 + 16x^5 - 13x^4 - 30x^3 - 38x^2 - 28x - 11.$$

Entnommen ist diese Kurve aus [Web97], Anhang B, Tabelle 8.

| $p = 4294972663$ |
|---|
| $\chi_{T_p}(x) = x^4 - 10848x^3 - 5405333827x^2 - 67429959513872x + 619650860194747773$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 48447869189186503 \cdot 305378179847563488733$ |

| $p = 4294973791$ |
|---|
| $\chi_{T_p}(x) = x^4 + 103360x^3 + 1095556521x^2 - 81235238473932x + 604525307712369793$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 7 \cdot 23 \cdot 73301837579 \cdot 28834457558899763875795651$ |

| $p = 4294974919$ |
|---|
| $\chi_{T_p}(x) = x^4 - 110180x^3 - 7678402016x^2 + 672059805968896x + 19026196475277617408$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 23 \cdot 59 \cdot 979516089957006308238184017332909$ |

| $p = 4294978021$ |
|---|
| $\chi_{T_p}(x) = x^4 + 78390x^3 - 5523623233x^2 - 326659472240122x - 1650022943773779227$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 7 \cdot 23 \cdot 1277 \cdot 26044397 \cdot 12702534991 \cdot 5002996852790947$ |

| $p = 4294978867$ |
|---|
| $\chi_{T_p}(x) = x^4 - 84300x^3 - 2671797855x^2 + 276696792946844x - 3909122412646827491$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3^2 \cdot 23 \cdot 1093 \cdot 150399050263457697347284722624 6007$ |

| $p = 4294981969$ |
|---|
| $\chi_{T_p}(x) = x^4 + 62798x^3 - 7503605373x^2 + 8704272645518x + 1365234976623409597$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 347 \cdot 138461 \cdot 39631769 \cdot 7770044342758031024533$ |

| $p = 4294983661$ |
|---|
| $\chi_{T_p}(x) = x^4 + 116164x^3 - 6977381680x^2 - 976302198770832x - 12896318070690012496$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 7 \cdot 23^2 \cdot 499 \cdot 719388659550948916777668558529$ |

| $p = 4294986763$ |
|---|
| $\chi_{T_p}(x) = x^4 + 112632x^3 - 4512573639x^2 - 284897214666932x + 8426841109973003169$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3^3 \cdot 23 \cdot 54798302783069298019371905764753578 1$ |

| $p = 4294989583$ |
|---|
| $\chi_{T_p}(x) = x^4 - 263528x^3 + 17000850045x^2 + 447640056346016x - 51245517060072766131$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 34848376847 \cdot 42453296782423162330242459 7$ |

| $p = 4294990429$ |
|---|
| $\chi_{T_p}(x) = x^4 - 44562x^3 - 5233033937x^2 - 116041611908986x - 756564771782208403$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 61 \cdot 24254181590766710131978967181605923 9$ |

| $p = 4294981321$ |
|---|
| $\chi_{T_p}(x) = x^4 + 123538x^3 - 3829593985x^2 - 488011193619218x + 12907283915109769793$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 19 \cdot 23 \cdot 8783 \cdot 79451 \cdot 37197407059529886868041349 9$ |

| $p = 4294982449$ |
|---|
| $\chi_{T_p}(x) = x^4 - 97934x^3 - 2701133425x^2 + 308838066985722x - 4690224614642381171$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 94727 \cdot 52061034918778872297247828749983$ |

| **Beispiele zur Stufe $N = 47$ (Fortsetzung)** |
|---|
| $p = 4294983013$ |
| $\chi_{T_p}(x) = x^4 - 124522x^3 + 5597019175x^2 - 106732626156470x + 720906954635207409$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 61 \cdot 3391 \cdot 276869 \cdot 5530749173 \cdot 46707804645574477$ |
| $p = 4294985269$ |
| $\chi_{T_p}(x) = x^4 - 217132x^3 + 14874575968x^2 - 357396699595472x + 2184696701595295344$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 23 \cdot 89 \cdot 233663 \cdot 770101 \cdot 3608521651154175872119$ |
| $p = 4294989781$ |
| $\chi_{T_p}(x) = x^4 - 196066x^3 + 10948408727x^2 - 86129034365826x - 4732631600075547787$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 37 \cdot 1318477339 \cdot 4506731997331 \cdot 22430734603021$ |
| $p = 4294996267$ |
| $\chi_{T_p}(x) = x^4 + 184636x^3 + 5489262349x^2 - 279055617416816x + 2286011891316874601$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 131371 \cdot 6662741 \cdot 471047823331 \cdot 35885959134419$ |
| $p = 4294996549$ |
| $\chi_{T_p}(x) = x^4 + 181178x^3 + 11619144395x^2 + 309429400954506x + 2890617850781495357$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3^3 \cdot 19 \cdot 23 \cdot 131 \cdot 199 \cdot 19345502171659 \cdot 57189989442441733$ |
| $p = 4295000779$ |
| $\chi_{T_p}(x) = x^4 + 150228x^3 + 5123558733x^2 - 88987771113832x - 4333763314971009839$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 67869467 \cdot 155247193 \cdot 468080870424776370959$ |
| $p = 4295001343$ |
| $\chi_{T_p}(x) = x^4 - 73596x^3 - 12448676971x^2 + 804938378199016x + 16582954548026030521$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 49317003774927218856343033605764354 29$ |
| $p = 42950027539$ |
| $\chi_{T_p}(x) = x^4 - 93794x^3 + 440096907x^2 + 83945759880294x + 651378480389287029$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 18334866857598882907 \cdot 61866254363127923730509$ |
| $p = 4294972093$ |
| $\chi_{T_p}(x) = x^4 - 49304x^3 - 13014483944x^2 + 497351124232672x - 4150023601883964016$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 23 \cdot 2837 \cdot 4545473 \cdot 4481569958732150758486981$ |
| $p = 4294972657$ |
| $\chi_{T_p}(x) = x^4 + 33064x^3 - 11930357668x^2 - 79462603892208x + 18328332474338654704$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3^2 \cdot 23 \cdot 103 \cdot 131 \cdot 2007537813646789 \cdot 3792995906349323$ |
| $p = 4294974913$ |
| $\chi_{T_p}(x) = x^4 + 44246x^3 - 12536912524x^2 - 585519312770472x + 14908831879836897008$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3 \cdot 23 \cdot 61 \cdot 11349052591 \cdot 44523416553054462061 8109$ |
| $p = 4294976887$ |
| $\chi_{T_p}(x) = x^4 + 192320x^3 + 12083737284x^2 + 306215070673504x + 2717290123756155408$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 97 \cdot 821 \cdot 11149 \cdot 10415116825716886749923432 23$ |
| $p = 4294983937$ |
| $\chi_{T_p}(x) = x^4 + 106688x^3 - 4669094160x^2 - 482760826268472x - 2272887379020158928$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 19 \cdot 23^2 \cdot 163 \cdot 487 \cdot 2657 \cdot 549316841 \cdot 18264032626396559$ |
| $p = 4294984501$ |
| $\chi_{T_p}(x) = x^4 - 171696x^3 + 5478021416x^2 + 191702419382504x - 7381638341878895472$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 15013 \cdot 615905013719518061841058078054 97$ |

| **Beispiele zur Stufe $N = 47$ (Fortsetzung)** |
|---|
| $p = 4294985911$ |
| $\chi_{T_p}(x) = x^4 + 245508x^3 + 18208931600x^2 + 423073322352448x + 1928034833265033472$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 3^3 \cdot 7 \cdot 23 \cdot 59 \cdot 97 \cdot 131 \cdot 199 \cdot 677 \cdot 3027648175507728902531$ |
| $p = 4294986193$ |
| $\chi_{T_p}(x) = x^4 - 128372x^3 - 9392676408x^2 + 1813846016376200x - 60607238382318250384$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3^3 \cdot 23 \cdot 5309 \cdot 645074388670911212957614284 2449$ |
| $p = 4294986757$ |
| $\chi_{T_p}(x) = x^4 + 8876x^3 - 5506898720x^2 - 154708998194816x + 237369527651380992$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 283 \cdot 292676101891 \cdot 11164176602079958253773$ |
| $p = 4294990423$ |
| $\chi_{T_p}(x) = x^4 + 20954x^3 - 6365210400x^2 + 150628440347712x - 302002013853546768$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23^3 \cdot 48583952801 \cdot 35979455055381598443161$ |
| $p = 4294975393$ |
| $\chi_{T_p}(x) = x^4 + 65650x^3 - 56911209x^2 - 23293485936558x - 89355802458711323$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 7 \cdot 23 \cdot 1451 \cdot 9551 \cdot 3682543 \cdot 41415165236647580078639$ |
| $p = 4294978777$ |
| $\chi_{T_p}(x) = x^4 + 228850x^3 + 14190612351x^2 + 144348018903258x - 3714592835043400163$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 78643307 \cdot 21889016843 \cdot 8595108638803045187$ |
| $p = 4294980751$ |
| $\chi_{T_p}(x) = x^4 - 76128x^3 - 1055479723x^2 + 119084915691832x - 444206405966385371$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 7 \cdot 23 \cdot 704514699295823475089094340990573271$ |
| $p = 4294982443$ |
| $\chi_{T_p}(x) = x^4 - 8496x^3 - 9602957387x^2 + 750438214102 16x + 1394054116237848677$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 15913 \cdot 92974706005323353981391092206 5019$ |
| $p = 4294984699$ |
| $\chi_{T_p}(x) = x^4 + 160732x^3 + 9129966993x^2 + 214614151315076x + 1780164971076207141$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 7 \cdot 23 \cdot 89 \cdot 1657 \cdot 1433257905243128989561927788 4597$ |
| $p = 4294985263$ |
| $\chi_{T_p}(x) = x^4 - 79872x^3 - 7337695824x^2 + 775597669562752x - 15580593745297174784$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 7 \cdot 23 \cdot 6918985848496507 \cdot 1193247002708314949$ |
| $p = 4294989211$ |
| $\chi_{T_p}(x) = x^4 - 47776x^3 - 1797369899x^2 + 92999668626376x - 787597411626061563$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 7 \cdot 23 \cdot 248390103433 \cdot 8509093854771257964600877$ |
| $p = 4294990621$ |
| $\chi_{T_p}(x) = x^4 + 66x^3 - 12606244393x^2 - 268304783821438x + 699757368527536981$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 1061 \cdot 139445873164050111172935252477763 19$ |
| $p = 4294991749$ |
| $\chi_{T_p}(x) = x^4 - 157608x^3 - 1290942920x^2 + 997590869244448x - 29327240216580946672$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 3 \cdot 23 \cdot 43 \cdot 59 \cdot 61299131 \cdot 87287617 \cdot 1419112810361423$ |
| $p = 4294993723$ |
| $\chi_{T_p}(x) = x^4 - 23376x^3 - 8445078016x^2 + 165908600931136x + 782360184251681536$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 3 \cdot 19 \cdot 23 \cdot 10847 \cdot 134339 \cdot 695814319455227162334337$ |

| **Beispiele zur Stufe $N = 47$ (Fortsetzung)** |
|---|
| $p = 4294977559$ |
| $\chi_{T_p}(x) = x^4 - 311544x^3 + 31940707760x^2 - 1196226912580736x + 14517629642861428992$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 23 \cdot 467 \cdot 37020755782793 \cdot 3342582469463707669$ |
| $p = 4294978123$ |
| $\chi_{T_p}(x) = x^4 + 126984x^3 - 3553594239x^2 - 978377514047860x - 31449501829842371999$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 347 \cdot 67493 \cdot 2023533427 \cdot 104066029147168165993$ |
| $p = 4294986019$ |
| $\chi_{T_p}(x) = x^4 + 265140x^3 + 23746448240x^2 + 878335269733568x + 11549130977846893824$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 7 \cdot 23 \cdot 631 \cdot 148036403 \cdot 88391418484867688071393$ |
| $p = 4294988557$ |
| $\chi_{T_p}(x) = x^4 + 75054x^3 - 4490210905x^2 - 69821937072466x + 1085551029300150453$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 203183 \cdot 86885087 \cdot 5503255777 \cdot 152291439199139$ |
| $p = 4294992223$ |
| $\chi_{T_p}(x) = x^4 - 82976x^3 - 4490346231x^2 - 15295629043116x + 169504898708533441$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 3454602491792993 \cdot 4282675994823235734727$ |
| $p = 4294996171$ |
| $\chi_{T_p}(x) = x^4 - 120312x^3 - 3897504747x^2 + 877133057286232x - 25404672439642319363$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 773 \cdot 135029 \cdot 3568951 \cdot 650775319 \cdot 61028531270387$ |
| $p = 4294998709$ |
| $\chi_{T_p}(x) = x^4 + 23194x^3 - 11185034105x^2 - 196060769468558x + 5101665126890103149$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 487 \cdot 5471 \cdot 50162568637 \cdot 110700916625517898247$ |
| $p = 4295000119$ |
| $\chi_{T_p}(x) = x^4 - 10036x^3 - 10272490683x^2 - 488728923404456x - 6206990629764351423$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 14795303219106301210010447314761149159$ |
| $p = 4295002093$ |
| $\chi_{T_p}(x) = x^4 + 44726x^3 - 4063088665x^2 - 114145731872482x - 573807228781007203$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 59 \cdot 319427 \cdot 1386592187 \cdot 566184300636624207553$ |
| $p = 4295002939$ |
| $\chi_{T_p}(x) = x^4 + 196840x^3 + 11212567073x^2 + 193289988727564x + 140280527403172553$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 353 \cdot 521 \cdot 83659001 \cdot 320552797724413222387829$ |
| $p = 4294969177$ |
| $\chi_{T_p}(x) = x^4 + 197902x^3 - 1495711600x^2 - 1985108580509744x - 82090667221334952976$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3 \cdot 7 \cdot 17^2 \cdot 23 \cdot 602422741 \cdot 2133644771 \cdot 118541873636621$ |
| $p = 4294971151$ |
| $\chi_{T_p}(x) = x^4 - 282204x^3 + 25544822048x^2 - 725007396095360x - 11622013692626040 32$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 3 \cdot 23 \cdot 37 \cdot 719 \cdot 937 \cdot 7207 \cdot 107225649411740089707233$ |
| $p = 4294973407$ |
| $\chi_{T_p}(x) = x^4 + 87544x^3 - 1473475072x^2 - 139609305207744x - 1052584945928558336$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 3 \cdot 23 \cdot 27739 \cdot 14990959 \cdot 106193851 \cdot 436256716069627$ |
| $p = 4294978201$ |
| $\chi_{T_p}(x) = x^4 + 59392x^3 - 1430909248x^2 - 62065290275000x + 532719341926766256$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 7^2 \cdot 23 \cdot 188714800749746027935103465108006 63$ |

| **Beispiele zur Stufe $N = 47$ (Fortsetzung)** |
|---|
| $p = 4294979047$ |
| $\chi_{T_p}(x) = x^4 - 21162x^3 - 6114444076x^2 + 1933221198232x + 632716665396310864$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3^2 \cdot 23 \cdot 10274287878646067113302573613645255 9$ |
| $p = 4294979329$ |
| $\chi_{T_p}(x) = x^4 + 29658x^3 - 8100417544x^2 - 313734999216640x + 3961853502427864272$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 409 \cdot 99859 \cdot 22640654206544405975533430909$ |
| $p = 4294981021$ |
| $\chi_{T_p}(x) = x^4 - 129784x^3 - 1415137480x^2 + 486023962009560x - 8370846657610317776$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 59 \cdot 401 \cdot 711329 \cdot 5183557 \cdot 210253429 \cdot 50413423367$ |
| $p = 4294982431$ |
| $\chi_{T_p}(x) = x^4 - 89194x^3 - 1694363536x^2 + 142155712199040x + 2819680702746133232$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3 \cdot 23 \cdot 73 \cdot 42222565137168987666046928076681 71$ |
| $p = 4294983841$ |
| $\chi_{T_p}(x) = x^4 - 8934x^3 - 11820332804x^2 - 434042158433768x + 450933519562289744$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3 \cdot 23 \cdot 1031 \cdot 29896303586755573143467230322003 9$ |
| $p = 4294988353$ |
| $\chi_{T_p}(x) = x^4 + 9486x^3 - 6365287000x^2 - 34502037638432x + 9924065842953467344$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3 \cdot 23^2 \cdot 59 \cdot 61 \cdot 23142563 \cdot 732159199 \cdot 219762297754931$ |
| $p = 4294972567$ |
| $\chi_{T_p}(x) = x^4 - 163016x^3 + 6287139413x^2 + 61417118540160x - 4733429674987302899$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 47 \cdot 2339 \cdot 90874698121 \cdot 493633860094672807621$ |
| $p = 4294976797$ |
| $\chi_{T_p}(x) = x^4 + 292970x^3 + 29983154415x^2 + 1227674210770158x + 154846173429415328 89$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3^2 \cdot 23 \cdot 61 \cdot 373 \cdot 37835735618059 \cdot 1909684915948112101$ |
| $p = 4294977079$ |
| $\chi_{T_p}(x) = x^4 - 132796x^3 - 7081919551x^2 + 1197202852229556x - 19371139648619592707$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 31 \cdot 347 \cdot 647 \cdot 5876323 \cdot 190772611 \cdot 1896203769820273$ |
| $p = 4294978489$ |
| $\chi_{T_p}(x) = x^4 + 28718x^3 - 674155873x^2 + 3897837913910x - 6745338595506787$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 7 \cdot 23 \cdot 59 \cdot 35823580348956458531852060738272687$ |
| $p = 4294979053$ |
| $\chi_{T_p}(x) = x^4 + 24862x^3 - 2379958581x^2 + 17776797877682x + 264228832305520257$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 73 \cdot 211845341925961 \cdot 9567029196743723021 27$ |
| $p = 4294979617$ |
| $\chi_{T_p}(x) = x^4 + 86924x^3 - 1114265744x^2 - 125871645536176x + 1072732120958145584$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 3^2 \cdot 23 \cdot 3735817 \cdot 7990427 \cdot 2151232375123849674 43$ |
| $p = 4294981591$ |
| $\chi_{T_p}(x) = x^4 + 39572x^3 - 4851586911x^2 + 37373373152852x + 1418556528906124957$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 19 \cdot 23^2 \cdot 163 \cdot 81647 \cdot 7143067 \cdot 3561454461139194271 37$ |
| $p = 4294984411$ |
| $\chi_{T_p}(x) = x^4 - 75428x^3 - 871303075x^2 + 134529499406328x - 1842589328886868207$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 4799 \cdot 29696383 \cdot 152561685623 \cdot 680473348963217$ |

| **Beispiele zur Stufe $N = 47$ (Fortsetzung)** |
|---|
| $p = 4294986103$ |
| $\chi_{T_p}(x) = x^4 - 44036x^3 - 13334834512x^2 + 377003355137728x + 31767417065966686976$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 3^4 \cdot 7 \cdot 23 \cdot 2754701910439 \cdot 37001312428578885767$ |
| $p = 4294987231$ |
| $\chi_{T_p}(x) = x^4 + 59608x^3 - 6622349011x^2 - 480569722869944x - 56173233205666122 67$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 19777 \cdot 31193 \cdot 2977013970323 \cdot 2685381628396483$ |
| $p = 4294974451$ |
| $\chi_{T_p}(x) = x^4 - 93744x^3 - 6536391983x^2 + 734362794478036x - 10257992957642278327$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3^2 \cdot 7 \cdot 23 \cdot 234835891946837019609376907914327873$ |
| $p = 4294975297$ |
| $\chi_{T_p}(x) = x^4 + 100326x^3 + 1042486751x^2 - 94566493097482x - 985164974563761227$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 6763 \cdot 2511601 \cdot 871033694614619305295787601$ |
| $p = 4294978399$ |
| $\chi_{T_p}(x) = x^4 - 96764x^3 - 1077929311x^2 + 86143604025756x + 1232436233669715389$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 31 \cdot 83 \cdot 3313 \cdot 9769 \cdot 5358041357741 \cdot 33158045439083$ |
| $p = 4294979527$ |
| $\chi_{T_p}(x) = x^4 + 14528x^3 - 8936133611x^2 - 205082191650416x - 225890386637145891$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 2707 \cdot 5897 \cdot 58341307 \cdot 32323553491 \cdot 491477214881$ |
| $p = 4294979809$ |
| $\chi_{T_p}(x) = x^4 + 84010x^3 + 367714515x^2 - 74578803382414x - 370692926633072091$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 14563 \cdot 92075663 \cdot 889863407 \cdot 12399555388799441$ |
| $p = 4294981783$ |
| $\chi_{T_p}(x) = x^4 + 53264x^3 + 392842656x^2 - 1177427765184x - 4973965117323008$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 3 \cdot 23 \cdot 2789 \cdot 690737636863061811 6136552839359$ |
| $p = 4294984321$ |
| $\chi_{T_p}(x) = x^4 + 18826x^3 - 6040878013x^2 - 93264996215738x + 4016302707496434977$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 19 \cdot 23 \cdot 6528583 \cdot 67520093 \cdot 588834952986548953717$ |
| $p = 4294985449$ |
| $\chi_{T_p}(x) = x^4 - 9854x^3 - 10454641025x^2 + 81982287760318x + 5514251578037682737$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 225326240603 \cdot 656608021680506517706861 73$ |
| $p = 4294986013$ |
| $\chi_{T_p}(x) = x^4 + 110130x^3 - 628079793x^2 - 310752635093398x - 5740758705758715059$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 61 \cdot 137 \cdot 163 \cdot 471137 \cdot 768464557481915718155 1199$ |
| $p = 4294987141$ |
| $\chi_{T_p}(x) = x^4 + 154970x^3 + 4585807055x^2 - 822467863078x - 837594912557290651$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 67 \cdot 357583 \cdot 617566371967074969511333025783$ |
| $p = 4294976617$ |
| $\chi_{T_p}(x) = x^4 - 67126x^3 - 7307046828x^2 + 236770317755176x + 15194900132507468304$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 31 \cdot 298281909845041669662793914412241 93$ |
| $p = 4294978027$ |
| $\chi_{T_p}(x) = x^4 + 262034x^3 + 19195100292x^2 + 66569450454216x - 24045062192000533616$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3^2 \cdot 7 \cdot 23 \cdot 286541 \cdot 5122655389840507855732 1126111$ |

| **Beispiele zur Stufe $N = 47$ (Fortsetzung)** |
|---|
| $p = 4294978309$ |
| $\chi_{T_p}(x) = x^4 - 146810x^3 - 7307777404x^2 + 1072877583988792x + 22805494117657473008$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3^3 \cdot 23 \cdot 811 \cdot 2297 \cdot 183838142042153406881278709 69$ |
| $p = 4294978591$ |
| $\chi_{T_p}(x) = x^4 - 117232x^3 - 6900466240x^2 + 1126796634158808x - 29639986507129221104$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 97 \cdot 327866291 \cdot 290747399317264740710285 57$ |
| $p = 4294978873$ |
| $\chi_{T_p}(x) = x^4 - 186832x^3 + 11925624920x^2 - 285290603647336x + 1548877660817138864$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 73 \cdot 238151 \cdot 53186590095684687375589589977$ |
| $p = 4294979719$ |
| $\chi_{T_p}(x) = x^4 + 183014x^3 + 10938674096x^2 + 248705700153984x + 1673969108011660144$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 347 \cdot 2467 \cdot 73369 \cdot 14723258358311405317860953$ |
| $p = 4294980847$ |
| $\chi_{T_p}(x) = x^4 + 200776x^3 + 9580591320x^2 - 73801455158616x - 7246290995444932688$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 173 \cdot 1069 \cdot 31090858027 \cdot 160827715607196537751$ |
| $p = 4294981129$ |
| $\chi_{T_p}(x) = x^4 + 106520x^3 - 774962536x^2 - 102823558382560x - 1033466384002379376$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 23 \cdot 73 \cdot 8798777359 \cdot 8997934369849996396 5139$ |
| $p = 4294987051$ |
| $\chi_{T_p}(x) = x^4 - 6322x^3 - 9665042680x^2 - 56021441660752x + 12807806074484548656$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 23 \cdot 59 \cdot 157 \cdot 57679 \cdot 17307310739213673352651 47377$ |
| $p = 4294988179$ |
| $\chi_{T_p}(x) = x^4 + 196068x^3 + 11482342956x^2 + 161411982012272x - 2315483211588407024$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^4 \cdot 3^2 \cdot 23 \cdot 7369 \cdot 507607 \cdot 27468897596456341894668841$ |
| $p = 4294981417$ |
| $\chi_{T_p}(x) = x^4 + 175054x^3 + 9371925635x^2 + 1815738638265 54x + 960926022287086633$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 47 \cdot 10493392288002086260792500566578 9643$ |
| $p = 4294985083$ |
| $\chi_{T_p}(x) = x^4 - 39792x^3 - 959997267x^2 + 12884609505528x + 9914423932294749$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 3803 \cdot 351121 \cdot 23086534503581 \cdot 479924480585093$ |
| $p = 4294985647$ |
| $\chi_{T_p}(x) = x^4 - 77748x^3 - 2045961359x^2 + 48955672874668x + 844443428813922101$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 49316235216128555699139 09830900630641$ |
| $p = 4294986211$ |
| $\chi_{T_p}(x) = x^4 - 65580x^3 - 368158759x^2 + 18634222368788x - 96700450668621323$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 1193 \cdot 127399837 \cdot 487779144737 \cdot 199563027890263$ |
| $p = 4294987057$ |
| $\chi_{T_p}(x) = x^4 + 153258x^3 + 2227091155x^2 - 546484559720570x - 20513837408638256671$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3^2 \cdot 23 \cdot 16439650838182314174454632 50347529783$ |
| $p = 4294987621$ |
| $\chi_{T_p}(x) = x^4 + 64504x^3 - 8695436840x^2 - 473165211853600x + 9015574363455811344$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 23 \cdot 1481353 \cdot 92258217029137 \cdot 422885474922499$ |

| **Beispiele zur Stufe $N = 47$ (Fortsetzung)** |
|---|
| $p = 4294987903$ |
| $\chi_{T_p}(x) = x^4 + 240448x^3 + 20329344089x^2 + 723394158708052x + 9199020856331516673$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 47 \cdot 278022085037 \cdot 113231459018326535303738 9$ |
| $p = 4294989313$ |
| $\chi_{T_p}(x) = x^4 + 73142x^3 - 4900555585x^2 - 69915278551758x + 2437547507243362841$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 3 \cdot 23 \cdot 401 \cdot 131959 \cdot 397939 \cdot 30684659 \cdot 7632827746359323$ |
| $p = 4294989877$ |
| $\chi_{T_p}(x) = x^4 + 196502x^3 + 7744760963x^2 - 152626701751254x - 1042256887116411983$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 23 \cdot 53 \cdot 7690157 \cdot 1386311239 \cdot 3789070091 \cdot 6910924571$ |
| $p = 4294990723$ |
| $\chi_{T_p}(x) = x^4 + 100596x^3 - 8498076496x^2 - 1124986910790464x - 259455544693869770 24$ |
| $\#J_0(47)_p(\mathbb{F}_p) = 2^8 \cdot 3 \cdot 23 \cdot 67746193 \cdot 154096903 \cdot 1845403036540231279$ |

## A.3 Beispiele zur Stufe $N = 53$

Die Modulkurve $X_0(53)$ ist eine bielliptische Kurve von Geschlecht $g = 4$. Der Quotient von $X_0(53)$ mit der Atkin-Lehner-Involution $w_{53}$ ist die elliptische Kurve

$$y^2 - xy - y = x^3 - x^2.$$

$X_0(53)$ ist nicht hyperelliptisch und nach N. Elkies wird der Funktionenkörper von $X_0(53)$ dargestellt von $x$, $y$ und einer Wurzel von $f(x, y)$ mit

$$f(x, y) := x^4 - 7x^3 + 9x^2 - 8x - 11 - (2x^2 + 3x - 11)y.$$

| $p = 4294977667$ |
|---|
| $\chi_{T_p}(x) = x^4 - 146156x^3 + 1702231536x^2 + 249171671317892x + 668444913992480864$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 3^3 \cdot 13 \cdot 149 \cdot 113467 \cdot 266897 \cdot 6967717 \cdot 1927133166078019$ |

| $p = 4294980847$ |
|---|
| $\chi_{T_p}(x) = x^4 - 70212x^3 - 5250622852x^2 + 253009586956628x + 8236144080786843352$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 5^2 \cdot 13^2 \cdot 19 \cdot 29 \cdot 37 \cdot 67 \cdot 163 \cdot 23911 \cdot 7469872073 \cdot 253159653463$ |

| $p = 4294983073$ |
|---|
| $\chi_{T_p}(x) = x^4 + 129012x^3 - 3597231346x^2 - 611602620501784x + 34726107560367931 65$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 107 \cdot 40139977 \cdot 3123989929 \cdot 1950944869971175739$ |

| $p = 4294984663$ |
|---|
| $\chi_{T_p}(x) = x^4 - 22088x^3 - 13860548480x^2 + 231660030550816x - 277719490650427264$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^7 \cdot 13 \cdot 22573 \cdot 47569 \cdot 230273 \cdot 827055381213165432191$ |

| $p = 4294986889$ |
|---|
| $\chi_{T_p}(x) = x^4 - 139224x^3 - 288777758x^2 + 333228691338736x - 5889195209215366075$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3^2 \cdot 5 \cdot 13 \cdot 829 \cdot 1499 \cdot 115133 \cdot 995463151 \cdot 4084098924803993$ |

| $p = 4294991023$ |
|---|
| $\chi_{T_p}(x) = x^4 - 18600x^3 - 6017511776x^2 - 121926551350960x - 627697152030602992$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 13 \cdot 17^2 \cdot 137 \cdot 317 \cdot 811 \cdot 971 \cdot 8353 \cdot 19816346551394998537$ |

| $p = 4294991977$ |
|---|
| $\chi_{T_p}(x) = x^4 - 3820x^3 - 8049658978x^2 + 70040906706180x + 2912381497459419721$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 5^2 \cdot 13 \cdot 19 \cdot 2749 \cdot 82231 \cdot 24378207433699514550176 1797$ |

| $p = 4294994203$ |
|---|
| $\chi_{T_p}(x) = x^4 + 28696x^3 - 13010857380x^2 - 79503644094996x + 1826056598065186504$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 5 \cdot 13 \cdot 41 \cdot 73 \cdot 143501 \cdot 8838481 \cdot 987929249 \cdot 174495714551$ |

| $p = 4294995157$ |
|---|
| $\chi_{T_p}(x) = x^4 + 184536x^3 + 5585128226x^2 - 374099105171776x - 15872341523142450059$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 5^2 \cdot 13 \cdot 17519 \cdot 49033 \cdot 3096727 \cdot 18844261417 \cdot 20888448961$ |

| $p = 4294997383$ |
|---|
| $\chi_{T_p}(x) = x^4 + 226244x^3 + 17539024896x^2 + 563275543132080x + 6367009236359685952$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^6 \cdot 3 \cdot 5 \cdot 13^2 \cdot 409 \cdot 875107 \cdot 140654963 \cdot 41665541179353701$ |

| **Beispiele zur Stufe $N = 53$ (Fortsetzung)** |
|---|
| $p = 4294969177$ |
| $\chi_{T_p}(x) = x^4 + 102340x^3 - 7848775234x^2 - 731470291025084x - 341709023402878903$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3 \cdot 5^3 \cdot 13 \cdot 17 \cdot 41 \cdot 547 \cdot 751 \cdot 26171 \cdot 9315272745292222756381$ |
| $p = 4294969813$ |
| $\chi_{T_p}(x) = x^4 + 122092x^3 - 5011131850x^2 - 792523527313844x - 17649459968783383663$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3 \cdot 5 \cdot 13 \cdot 449 \cdot 3188483 \cdot 3626153 \cdot 199023949 \cdot 1689025812917$ |
| $p = 4294972039$ |
| $\chi_{T_p}(x) = x^4 + 165428x^3 + 10202821056x^2 + 277990433324784x + 2822856109869634240$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^6 \cdot 3^2 \cdot 5^2 \cdot 13 \cdot 19 \cdot 359 \cdot 17491 \cdot 120211193 \cdot 126749031740375467$ |
| $p = 4294973629$ |
| $\chi_{T_p}(x) = x^4 - 43644x^3 - 2460861058x^2 + 153619851019204x - 2003695444213020935$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3 \cdot 5 \cdot 13 \cdot 59 \cdot 269 \cdot 13883 \cdot 18041 \cdot 831167 \cdot 271783037 \cdot 1943319109$ |
| $p = 4294974583$ |
| $\chi_{T_p}(x) = x^4 + 54140x^3 - 8535847752x^2 - 669395148972624x - 11086677806721092480$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^8 \cdot 3^2 \cdot 7 \cdot 13 \cdot 43 \cdot 151 \cdot 197 \cdot 503 \cdot 863 \cdot 2923032136419980807743$ |
| $p = 4294975537$ |
| $\chi_{T_p}(x) = x^4 - 88512x^3 - 6893375710x^2 + 316603519960388x + 15129184236667608345$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 7 \cdot 11 \cdot 13 \cdot 431 \cdot 569 \cdot 18233 \cdot 129419 \cdot 369013 \cdot 1591887513734489$ |
| $p = 4294977763$ |
| $\chi_{T_p}(x) = x^4 + 68388x^3 - 9675089264x^2 - 705553133176208x - 9225388768386033232$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 3 \cdot 5^2 \cdot 11 \cdot 13 \cdot 89 \cdot 233 \cdot 367 \cdot 761 \cdot 3347 \cdot 102301252115264783759$ |
| $p = 4294978081$ |
| $\chi_{T_p}(x) = x^4 + 125272x^3 + 774105218x^2 - 169015194971904x - 26292969813346555531$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3 \cdot 5 \cdot 7 \cdot 13 \cdot 541 \cdot 1847 \cdot 75611 \cdot 526871 \cdot 635737573 \cdot 9851276443$ |
| $p = 4294978399$ |
| $\chi_{T_p}(x) = x^4 + 123572x^3 + 1567752840x^2 - 135737230876832x + 1179546908681999024$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 3^3 \cdot 13 \cdot 17 \cdot 67^2 \cdot 107 \cdot 263 \cdot 28215729241584482743913233$ |
| $p = 4294980307$ |
| $\chi_{T_p}(x) = x^4 + 163432x^3 + 858017160x^2 - 545555042322172x - 8809318713968858776$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 3 \cdot 7 \cdot 11 \cdot 13 \cdot 19^2 \cdot 23 \cdot 404197 \cdot 199014923 \cdot 21208118801263949$ |
| $p = 4294971937$ |
| $\chi_{T_p}(x) = x^4 - 220310x^3 + 13338721100x^2 - 153820256923052x - 245955838905013840$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 3^2 \cdot 5^2 \cdot 13^2 \cdot 67 \cdot 3137 \cdot 5851 \cdot 241601 \cdot 3764798080877950741$ |
| $p = 4294975117$ |
| $\chi_{T_p}(x) = x^4 - 245866x^3 + 20963839632x^2 - 706091061705068x + 7303902141575654520$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 13 \cdot 137 \cdot 12143 \cdot 88423 \cdot 11120968348318317820586887$ |
| $p = 4294975753$ |
| $\chi_{T_p}(x) = x^4 + 19824x^3 - 5629600192x^2 - 111242754803248x + 5072022474980793280$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 13 \cdot 29 \cdot 67 \cdot 421 \cdot 175891 \cdot 973817027 \cdot 11676311085313481$ |
| $p = 4294976071$ |
| $\chi_{T_p}(x) = x^4 - 94796x^3 - 4814856814x^2 + 262192464534186x + 5254274311970211923$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 691 \cdot 2396683 \cdot 6215431 \cdot 2542905167282151748393$ |

| **Beispiele zur Stufe $N = 53$ (Fortsetzung)** |
|---|
| $p = 4294982431$ |
| $\chi_{T_p}(x) = x^4 - 40910x^3 - 4057372612x^2 + 184555924468180x - 1211262191908938000$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 11 \cdot 13 \cdot 37 \cdot 43 \cdot 2270071 \cdot 411789994212393900378682 01$ |
| $p = 4294986247$ |
| $\chi_{T_p}(x) = x^4 + 53738x^3 - 1573230570x^2 - 52389830346800x + 656806711282859375$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3^3 \cdot 7 \cdot 13 \cdot 23 \cdot 163 \cdot 179 \cdot 233 \cdot 1783 \cdot 12473 \cdot 163661 \cdot 243363548533643$ |
| $p = 4294988473$ |
| $\chi_{T_p}(x) = x^4 - 84854x^3 - 10235389008x^2 + 783068823042844x + 8972468304452741992$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^5 \cdot 3^4 \cdot 5 \cdot 13 \cdot 233 \cdot 178958279 \cdot 48437683024442650012807$ |
| $p = 4294990699$ |
| $\chi_{T_p}(x) = x^4 - 78636x^3 - 9999040450x^2 + 806134633566782x - 2682557152575208857$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13^2 \cdot 137 \cdot 330378131 \cdot 662204849359 \cdot 67178459234539$ |
| $p = 4294991653$ |
| $\chi_{T_p}(x) = x^4 - 61448x^3 + 215529360x^2 + 16046118161232x - 48065894466467456$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 7^2 \cdot 13 \cdot 29 \cdot 131 \cdot 334549 \cdot 26269671314607259470390073$ |
| $p = 4294994833$ |
| $\chi_{T_p}(x) = x^4 - 119140x^3 + 145639608x^2 + 300341417829840x - 5826712306214539584$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 5^2 \cdot 11 \cdot 13 \cdot 2683 \cdot 72763 \cdot 3047272117884938010382 4797$ |
| $p = 4294977559$ |
| $\chi_{T_p}(x) = x^4 - 15502x^3 - 10644719394x^2 - 344853090357700x - 2945912469642560969$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 7 \cdot 13^2 \cdot 151 \cdot 104761 \cdot 410903 \cdot 613570939 \cdot 72123469074611$ |
| $p = 4294980421$ |
| $\chi_{T_p}(x) = x^4 - 151960x^3 + 3295830488x^2 + 121133132086560x - 530630110070785264$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^8 \cdot 3^2 \cdot 13^2 \cdot 37 \cdot 41 \cdot 137 \cdot 277 \cdot 691 \cdot 4211 \cdot 1195567 \cdot 4363521134759$ |
| $p = 4294983601$ |
| $\chi_{T_p}(x) = x^4 - 90x^3 - 10974613304x^2 - 138349248719788x + 12104803624225109720$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^5 \cdot 13^2 \cdot 14561 \cdot 36871 \cdot 4819501 \cdot 24318184821420220163$ |
| $p = 4294986781$ |
| $\chi_{T_p}(x) = x^4 + 44014x^3 - 6152861868x^2 - 235991535016108x - 1223491064230952448$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 13^4 \cdot 37 \cdot 337 \cdot 1023541 \cdot 6372511 \cdot 18312216657399311$ |
| $p = 4294988689$ |
| $\chi_{T_p}(x) = x^4 - 117914x^3 + 4970575416x^2 - 88179290303332x + 558966698255723864$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 3 \cdot 7^2 \cdot 13^2 \cdot 1279 \cdot 5711 \cdot 11720049566212354654996 4797$ |
| $p = 4294990597$ |
| $\chi_{T_p}(x) = x^4 + 93238x^3 - 10282617636x^2 - 543716732459772x + 35115822711392835168$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 3^2 \cdot 5 \cdot 11 \cdot 13^3 \cdot 41 \cdot 105817 \cdot 19285812239 \cdot 467472232094653$ |
| $p = 4294991551$ |
| $\chi_{T_p}(x) = x^4 + 110454x^3 + 1971326988x^2 - 65400547716668x - 1468968056115777920$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^7 \cdot 13^2 \cdot 2657 \cdot 4057 \cdot 33083 \cdot 934121 \cdot 104553997 \cdot 451667173$ |
| $p = 4294993141$ |
| $\chi_{T_p}(x) = x^4 + 32748x^3 - 9165222160x^2 + 49900894641728x - 20965579448929280$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 11 \cdot 13^2 \cdot 19 \cdot 15319 \cdot 5332627 \cdot 10275091 \cdot 717369546688177$ |

| **Beispiele zur Stufe $N = 53$ (Fortsetzung)** |
|---|
| $p = 4294993777$ |
| $\chi_{T_p}(x) = x^4 - 13618x^3 - 5153403792x^2 - 139043391567492x - 1014171747546073520$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 3^2 \cdot 13^2 \cdot 19 \cdot 173 \cdot 229 \cdot 317 \cdot 4954409 \cdot 23656106861006189863$ |
| $p = 4294996003$ |
| $\chi_{T_p}(x) = x^4 - 135562x^3 + 4337063734x^2 + 15830093034628x - 280059038446701969$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 5^2 \cdot 13^2 \cdot 19 \cdot 151 \cdot 112120033 \cdot 858987319 \cdot 291480910741109$ |
| $p = 4294981063$ |
| $\chi_{T_p}(x) = x^4 + 116936x^3 - 3863162024x^2 - 796497499968220x - 21099514436616593248$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^6 \cdot 3 \cdot 7 \cdot 13 \cdot 19 \cdot 15383 \cdot 35831 \cdot 12782807 \cdot 145490425773068591$ |
| $p = 4294981381$ |
| $\chi_{T_p}(x) = x^4 + 164148x^3 + 2806423410x^2 - 429553522457792x - 12869733417341714231$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 5 \cdot 13 \cdot 59 \cdot 694591 \cdot 4295044609 \cdot 29744011566451136077$ |
| $p = 4294988377$ |
| $\chi_{T_p}(x) = x^4 + 259796x^3 + 22437113472x^2 + 724429647649072x + 7739147017191704304$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^10 \cdot 13^2 \cdot 2549 \cdot 8101 \cdot 97757617 \cdot 974155991750075117$ |
| $p = 4294989331$ |
| $\chi_{T_p}(x) = x^4 - 28440x^3 - 11564734360x^2 - 54396124998016x + 19419547427501773200$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 7 \cdot 13 \cdot 61 \cdot 5029333 \cdot 76180510726544929606 4373443$ |
| $p = 4294989649$ |
| $\chi_{T_p}(x) = x^4 + 34768x^3 - 2893694262x^2 - 131817984411400x - 1009623071221943187$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 107 \cdot 1693 \cdot 4294999733 \cdot 9204486463 \cdot 3655150043269$ |
| $p = 4294991557$ |
| $\chi_{T_p}(x) = x^4 - 54436x^3 - 2180966286x^2 + 717436111286 48x - 474092980385166935$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 2003 \cdot 10709 \cdot 401057 \cdot 552992327 \cdot 5502313451983981$ |
| $p = 4294994101$ |
| $\chi_{T_p}(x) = x^4 + 7716x^3 - 6230035250x^2 + 4671716611076x + 948344990064752729$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 5^5 \cdot 13 \cdot 43 \cdot 59 \cdot 15359 \cdot 68209 \cdot 279641 \cdot 11270203466012423$ |
| $p = 4294996327$ |
| $\chi_{T_p}(x) = x^4 + 59848x^3 - 10796283764x^2 - 388046620485020x - 127808371409514616$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 5^2 \cdot 13 \cdot 41 \cdot 919 \cdot 11399 \cdot 275357 \cdot 1106678945656750951469$ |
| $p = 4294997281$ |
| $\chi_{T_p}(x) = x^4 + 156180x^3 + 993650198x^2 - 518345067004816x - 10615207315104690675$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 11 \cdot 13 \cdot 839 \cdot 21521 \cdot 465383 \cdot 486821 \cdot 581737554132346751$ |
| $p = 4294998553$ |
| $\chi_{T_p}(x) = x^4 + 20432x^3 - 5712572326x^2 - 61127344151644x + 8345552105669020745$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13^2 \cdot 5021 \cdot 855419 \cdot 5678807 \cdot 22134667 \cdot 3729651333563$ |
| $p = 4294982551$ |
| $\chi_{T_p}(x) = x^4 - 248840x^3 + 21931092922x^2 - 802456243411430x + 10285712366806563947$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 17^2 \cdot 24223 \cdot 4294889251 \cdot 870561305268752735411$ |
| $p = 4294982869$ |
| $\chi_{T_p}(x) = x^4 + 63950x^3 - 8807482780x^2 - 580324524246484x - 2622303635170816480$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 5 \cdot 13 \cdot 2147539223 \cdot 30472477292446751401702 7789$ |

| **Beispiele zur Stufe $N = 53$ (Fortsetzung)** |
|---|
| $p = 4294986049$ |
| $\chi_{T_p}(x) = x^4 - 88004x^3 - 1651018024x^2 + 306940125337072x - 6400144256820497920$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 3^2 \cdot 5 \cdot 13 \cdot 563 \cdot 569 \cdot 3461 \cdot 83869 \cdot 390965609729772066481$ |
| $p = 4294994317$ |
| $\chi_{T_p}(x) = x^4 + 13644x^3 - 9272964600x^2 - 146584010306224x - 52289697987107456$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 3 \cdot 13 \cdot 715816177 \cdot 10571660839 \cdot 72064675214668043$ |
| $p = 4294996543$ |
| $\chi_{T_p}(x) = x^4 - 200900x^3 + 12601497994x^2 - 241265398517306x - 590162705531111389$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 379 \cdot 38239 \cdot 11332309 \cdot 159376198525580436296951$ |
| $p = 4294998769$ |
| $\chi_{T_p}(x) = x^4 - 186968x^3 + 12675368248x^2 - 369007383901696x + 3896423386117848912$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^7 \cdot 13 \cdot 43 \cdot 439 \cdot 18047 \cdot 59497 \cdot 516589 \cdot 19529976971000789$ |
| $p = 4295001313$ |
| $\chi_{T_p}(x) = x^4 + 124810x^3 - 2648754664x^2 - 752594092788996x - 21631441070250732848$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 3 \cdot 5 \cdot 13 \cdot 239 \cdot 715843517 \cdot 1275044166854112191545609$ |
| $p = 4295002903$ |
| $\chi_{T_p}(x) = x^4 - 93150x^3 - 8041142154x^2 + 569789177202260x + 8312370888422756623$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3 \cdot 5 \cdot 13 \cdot 257 \cdot 1114159 \cdot 609437986519736260179793963$ |
| $p = 4295003857$ |
| $\chi_{T_p}(x) = x^4 + 93180x^3 + 2855410880x^2 + 31700069304224x + 95924640938668240$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^1 1 \cdot 7 \cdot 13^2 \cdot 19 \cdot 139 \cdot 252293 \cdot 371545785727 \cdot 567365620637$ |
| $p = 4295006083$ |
| $\chi_{T_p}(x) = x^4 + 57406x^3 - 4103069524x^2 + 53476345099620x - 195328347920721728$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 3 \cdot 13 \cdot 73 \cdot 383 \cdot 4902967 \cdot 3978283641783991103023919$ |
| $p = 4294971829$ |
| $\chi_{T_p}(x) = x^4 - 96896x^3 + 2369213240x^2 + 9322030525200x - 491419566446642000$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^8 \cdot 3 \cdot 5^3 \cdot 13 \cdot 137 \cdot 823 \cdot 17895769 \cdot 95458019 \cdot 1415577694081$ |
| $p = 4294972147$ |
| $\chi_{T_p}(x) = x^4 + 20508x^3 - 8024923872x^2 - 39939112520928x + 1403612565052937616$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 13 \cdot 17 \cdot 61 \cdot 1043299 \cdot 2070911 \cdot 36669329 \cdot 19912610579261$ |
| $p = 4294973101$ |
| $\chi_{T_p}(x) = x^4 - 67544x^3 - 3842993480x^2 + 17745842515472x + 526826505075744432$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^6 \cdot 5 \cdot 13 \cdot 97 \cdot 2213923 \cdot 380896854631948723784365207$ |
| $p = 4294976281$ |
| $\chi_{T_p}(x) = x^4 + 72468x^3 - 138552102x^2 - 27827762519288x + 209772060108259585$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 23 \cdot 31 \cdot 823 \cdot 809383 \cdot 5218669 \cdot 10560971225653964729$ |
| $p = 4294979461$ |
| $\chi_{T_p}(x) = x^4 - 17612x^3 - 14343193442x^2 + 276638870862880x + 34968580994626480013$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 5^3 \cdot 13 \cdot 17 \cdot 50528017 \cdot 2308473850591 \cdot 105604660320707$ |
| $p = 4294980733$ |
| $\chi_{T_p}(x) = x^4 + 78120x^3 - 12779019304x^2 - 1049321312258656x - 9187572270621252336$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^1 4 \cdot 13 \cdot 17 \cdot 487 \cdot 3691 \cdot 7591 \cdot 35363 \cdot 56681 \cdot 413081 \cdot 8318533$ |

| **Beispiele zur Stufe $N = 53$ (Fortsetzung)** |
|---|
| $p = 4294986139$ |
| $\chi_{T_p}(x) = x^4 - 54180x^3 - 3802236672x^2 + 20209197298580x + 654644241516996400$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^5 \cdot 3 \cdot 5^2 \cdot 13 \cdot 19 \cdot 1087 \cdot 1733 \cdot 12648166634503 \cdot 24092269230851$ |
| $p = 4294993771$ |
| $\chi_{T_p}(x) = x^4 - 195472x^3 + 2917687408x^2 + 1064264158970464x - 38518501909862745600$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^7 \cdot 5 \cdot 13 \cdot 37 \cdot 67 \cdot 86629 \cdot 223061696317 \cdot 853772639571587$ |
| $p = 4294996633$ |
| $\chi_{T_p}(x) = x^4 - 228860x^3 + 13425472226x^2 + 169271899139328x - 22841820204240940559$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3^2 \cdot 5 \cdot 13 \cdot 3719 \cdot 128321 \cdot 12188447460887105227099689 83$ |
| $p = 4294997587$ |
| $\chi_{T_p}(x) = x^4 + 271744x^3 + 25509444312x^2 + 967038280129308x + 12206081550693140800$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 13 \cdot 19^3 \cdot 37 \cdot 4231 \cdot 313883 \cdot 840465491 \cdot 5775939110443$ |
| $p = 4294973953$ |
| $\chi_{T_p}(x) = x^4 + 125836x^3 + 4292671352x^2 + 19901740727776x - 45752572734240144$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^7 \cdot 11 \cdot 13 \cdot 17^2 \cdot 281 \cdot 601 \cdot 52532993 \cdot 2676309407 \cdot 2709329057$ |
| $p = 4294975543$ |
| $\chi_{T_p}(x) = x^4 - 11838x^3 - 4849572540x^2 - 81458118675028x - 250293948508269920$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^6 \cdot 3^2 \cdot 11 \cdot 13^2 \cdot 31 \cdot 47 \cdot 61 \cdot 487 \cdot 1861 \cdot 19577 \cdot 143933749 \cdot 1400123069$ |
| $p = 4294977769$ |
| $\chi_{T_p}(x) = x^4 + 101770x^3 - 1569676200x^2 - 151292218405500x - 724325777431325000$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^5 \cdot 5^4 \cdot 13 \cdot 31 \cdot 107374573 \cdot 393203595401967984812081$ |
| $p = 4294978087$ |
| $\chi_{T_p}(x) = x^4 - 13010x^3 - 10629540578x^2 + 52175302515116x + 26756317545079186323$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 5 \cdot 13 \cdot 23 \cdot 37 \cdot 1543 \cdot 121021 \cdot 150881 \cdot 218342541131783581507$ |
| $p = 4294981903$ |
| $\chi_{T_p}(x) = x^4 + 155162x^3 - 2229753472x^2 - 1206354331881204x - 43185273375559669856$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^5 \cdot 5^2 \cdot 13 \cdot 31 \cdot 37 \cdot 21247 \cdot 17318879 \cdot 77525913000869695493$ |
| $p = 4294983811$ |
| $\chi_{T_p}(x) = x^4 - 59806x^3 - 12812392810x^2 + 500691577005804x + 36781515660489058211$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3^3 \cdot 13^2 \cdot 29 \cdot 61 \cdot 3913181 \cdot 5485177 \cdot 1963998687768482933$ |
| $p = 4294985083$ |
| $\chi_{T_p}(x) = x^4 + 255650x^3 + 22755980408x^2 + 855605390188500x + 11621527913815492816$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^7 \cdot 5^2 \cdot 13 \cdot 19 \cdot 37 \cdot 67 \cdot 1312513 \cdot 21773669 \cdot 45278953 \cdot 134220059$ |
| $p = 4294986991$ |
| $\chi_{T_p}(x) = x^4 + 73500x^3 - 8717974494x^2 - 898126137840494x - 18986666572631342153$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 17 \cdot 179 \cdot 4295055979 \cdot 9797372581 \cdot 204423625340119$ |
| $p = 4294990171$ |
| $\chi_{T_p}(x) = x^4 + 65014x^3 - 8930548140x^2 - 192202850457172x + 17221445375077064592$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^5 \cdot 13 \cdot 157 \cdot 1063 \cdot 3217 \cdot 521551 \cdot 11498617177 \cdot 254059573769$ |
| $p = 4294991443$ |
| $\chi_{T_p}(x) = x^4 - 105224x^3 - 2020749142x^2 + 155793113209162x + 850859937331349103$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 7^2 \cdot 13^2 \cdot 31 \cdot 217499 \cdot 1697351 \cdot 3590584637561512430341$ |

| **Beispiele zur Stufe $N = 53$ (Fortsetzung)** |
|---|
| $p = 4294978729$ |
| $\chi_{T_p}(x) = x^4 - 105510x^3 - 2592877136x^2 + 133858622439724x + 2842373328128216576$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 3 \cdot 13 \cdot 23 \cdot 179 \cdot 173867 \cdot 12646411 \cdot 120479591690204702377$ |
| $p = 4294979047$ |
| $\chi_{T_p}(x) = x^4 - 59180x^3 - 4009463274x^2 + 340802297073998x - 5988187885491355609$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3 \cdot 5 \cdot 13 \cdot 107 \cdot 167 \cdot 47137 \cdot 114941 \cdot 8572699 \cdot 32688431 \cdot 64321297$ |
| $p = 4294981591$ |
| $\chi_{T_p}(x) = x^4 - 81948x^3 - 12877540066x^2 + 660010116444334x + 42816088313070338007$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 29 \cdot 73 \cdot 2028847 \cdot 62999392283 \cdot 96735765566558123$ |
| $p = 4294983181$ |
| $\chi_{T_p}(x) = x^4 + 194000x^3 + 10066731296x^2 + 104076719582192x - 2114779807408128880$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^6 \cdot 11 \cdot 13 \cdot 151 \cdot 757 \cdot 128951 \cdot 1364533 \cdot 1848706171440669337$ |
| $p = 4294987951$ |
| $\chi_{T_p}(x) = x^4 - 280132x^3 + 28448480098x^2 - 1244435791417366x + 19811672546791472375$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3 \cdot 5 \cdot 13^3 \cdot 37 \cdot 1877 \cdot 48859 \cdot 228953 \cdot 16548193 \cdot 803138917391$ |
| $p = 4294994311$ |
| $\chi_{T_p}(x) = x^4 + 53946x^3 - 8738956836x^2 - 378922505099620x - 2976542393331102416$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 3^2 \cdot 13 \cdot 61 \cdot 1955831 \cdot 1523662650434592103054922167$ |
| $p = 4294997173$ |
| $\chi_{T_p}(x) = x^4 - 263112x^3 + 23723908848x^2 - 788467374106096x + 4999499893920813840$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^7 \cdot 13 \cdot 59 \cdot 536863489 \cdot 956396453 \cdot 6750217470139291$ |
| $p = 4294997491$ |
| $\chi_{T_p}(x) = x^4 - 55010x^3 - 1436323524x^2 + 100845575955828x - 1117728364401638640$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^5 \cdot 3^3 \cdot 13 \cdot 23 \cdot 59 \cdot 14653 \cdot 122011 \cdot 1252777 \cdot 9968041688789699$ |
| $p = 4294997809$ |
| $\chi_{T_p}(x) = x^4 - 88684x^3 - 3264585664x^2 + 230645564228400x + 58754080201217837 92$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^9 \cdot 13 \cdot 31 \cdot 22727 \cdot 23623 \cdot 1266259 \cdot 668309891 \cdot 3629862853$ |
| $p = 4294999717$ |
| $\chi_{T_p}(x) = x^4 - 168986x^3 + 6284552848x^2 + 97179289488460x - 2087258215889852800$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 7 \cdot 13 \cdot 79 \cdot 229 \cdot 607 \cdot 2207 \cdot 8963 \cdot 2151776189818972041631$ |
| $p = 4294981807$ |
| $\chi_{T_p}(x) = x^4 + 6376x^3 - 9993788802x^2 - 181972881152314x + 13734044379924441807$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 5^2 \cdot 13 \cdot 43 \cdot 103^2 \cdot 75743 \cdot 404849 \cdot 534570121 \cdot 140016571423$ |
| $p = 4294982443$ |
| $\chi_{T_p}(x) = x^4 - 47574x^3 - 1717988742x^2 + 30560982854652x - 107160174744581637$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 317 \cdot 4294977173 \cdot 19225484213686769761392343$ |
| $p = 4294985623$ |
| $\chi_{T_p}(x) = x^4 - 255564x^3 + 19133960686x^2 - 285249829901434x - 7081105333447861189$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 257 \cdot 317 \cdot 3253 \cdot 1320301 \cdot 125991529 \cdot 593727160037339$ |
| $p = 4294987849$ |
| $\chi_{T_p}(x) = x^4 - 224326x^3 + 15737817228x^2 - 294008848459812x - 3591321045218068400$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^3 \cdot 3 \cdot 5^2 \cdot 11 \cdot 13 \cdot 65074627 \cdot 795858703 \cdot 76575550917696263$ |

| **Beispiele zur Stufe $N = 53$ (Fortsetzung)** |
|---|
| $p = 4294991983$ |
| $\chi_{T_p}(x) = x^4 + 132164x^3 - 1325795354x^2 - 401642732476414x - 7407480884692488337$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3 \cdot 5 \cdot 13 \cdot 67 \cdot 3929 \cdot 364379 \cdot 1819357358689390009764927 7$ |
| $p = 4294997071$ |
| $\chi_{T_p}(x) = x^4 + 137306x^3 + 5108042102x^2 + 26989521651240x - 335969540064889057$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 3 \cdot 5 \cdot 13^3 \cdot 31 \cdot 110128517 \cdot 3024701845792971521058071$ |
| $p = 4294997389$ |
| $\chi_{T_p}(x) = x^4 - 48156x^3 - 5004756048x^2 + 290140420991168x - 3698278303558748160$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 5 \cdot 13 \cdot 19 \cdot 709 \cdot 22604873 \cdot 107451257074335270191693 3$ |
| $p = 4294998343$ |
| $\chi_{T_p}(x) = x^4 + 63328x^3 - 11345360466x^2 - 704628692449350x - 10060652945837369213$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 13 \cdot 23 \cdot 37 \cdot 6343 \cdot 11807 \cdot 677107 \cdot 10351329581 \cdot 58600021979$ |
| $p = 4294998661$ |
| $\chi_{T_p}(x) = x^4 + 62730x^3 - 5042173080x^2 - 116703064534708x + 6170858490147530440$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 3^2 \cdot 5^2 \cdot 7 \cdot 13 \cdot 29 \cdot 757 \cdot 23509 \cdot 7267121 \cdot 276964531677571901$ |
| $p = 4294999297$ |
| $\chi_{T_p}(x) = x^4 - 31648x^3 - 8013256576x^2 - 8448849645024x + 234710835350019328$ |
| $\#J_0(53)_p(\mathbb{F}_p) = 2^4 \cdot 13 \cdot 263 \cdot 3779 \cdot 8165539 \cdot 9047485793 \cdot 22281295880071$ |

# Anhang B

# 43-Bit Beispiele zur Matrix des Heckeoperators

Hier führen wir einige Beispiele zur Matrix $B_{T_p}$ des Hecke-Operatores $T_p$ mit $p \sim 2^{43}$ prim zu den Modulkurven der Stufen $N = 47$ und 53 auf, erstellen das charakteristische Polynome $\chi_{T_p}$ und werten dieses in $\chi_{T_p}(p+1) = \#J_0(N)_p(\mathbb{F}_p)$ aus. Die zugrundeliegenden Algorithmen wurden in `C++` programmiert, wobei die Modulsymbolreduktion mit `Magma` erfolgte.

## B.1 Beispiele zur Stufe $N = 47$

Die Modulkurve $X_0(47)$ ist eine hyperelliptische Kurve von Geschlecht $g = 4$ mit reeller Multiplikation. Diese ist gegeben, durch die affine Gleichung

$$y^2 = x^{10} + 6x^9 + 11x^8 + 24x^7 + 19x^6 + 16x^5 - 13x^4 - 30x^3 - 38x^2 - 28x - 11.$$

Entnommen ist diese Kurve aus [Web97], Anhang B, Tabelle 8. Eine Basis für den zu den Spitzenformen gehörigen Raum der Modulsymbole $\mathcal{S}_2(47)$ ist gegeben durch die Modulsymbole

$$m_1 := \{-1/28, 0\} \quad m_2 := \{-1/35, 0\} \quad m_3 := \{-1/31, 0\} \quad m_4 := \{-1/23, 0\}.$$

**Beispiel B.1.1** Die Matrix des Hecke-Operators $T_p$ zu der 43 Bit großen Primzahl $p = 8796093027097$ ist

$$B_{T_p} = \begin{pmatrix} -2136278 & -426764 & 119260 & -1187176 \\ -927236 & -186678 & -1094060 & -2616896 \\ 1306436 & 379200 & -236254 & 1520824 \\ -165818 & -1201254 & -142036 & -186678 \end{pmatrix}.$$

Das charakteristische Polynom $\chi_{T_p}$ von $B_{T_p}$ ist

$$\begin{aligned} \chi_{T_p}(x) \quad = \quad & x^4 + 2745888x^3 - 1835697173864x^2 \\ & -7420787822715538048x - 3181686132148478207538544. \end{aligned}$$

Die Ordnung von $J_0(47)_p(\mathbb{F}_p)$ ist

$$\begin{aligned}\chi_{T_p}(p+1) &= 59863125885736215240413939656509260530354654256622 08\\ &= 2^8 \cdot 23 \cdot 1153\\ &\quad \cdot 88178413775465549524064614722741920489723544 7.\end{aligned}$$

Der letzte Faktor von $\#J_0(47)_p(\mathbb{F}_p)$ ist eine 150-Bit Primzahl.

**Beispiel B.1.2** Die Matrix des Hecke-Operators $T_p$ zu der 43 Bit großen Primzahl $p = 8796093025123$ ist

$$B_{T_p} = \begin{pmatrix} -3784452 & -4236453 & -2847761 & -5771750 \\ -184719 & -420648 & 1165859 & 2038508 \\ 2923989 & 2739270 & 2025412 & 3070594 \\ -1242087/2 & 2185113/2 & 127548 & -420648 \end{pmatrix}.$$

Das charakteristische Polynom $\chi_{T_p}$ von $B_{T_p}$ ist

$$\begin{aligned}\chi_{T_p}(x) &= x^4 + 2600336x^3 - 7860931654827x^2\\ &\quad -12546699278027633496x + 11511209415587362770738717.\end{aligned}$$

Die Ordnung von $J_0(47)_p(\mathbb{F}_p)$ ist

$$\begin{aligned}\chi_{T_p}(p+1) &= 59863124841418567016748967381086815598374496328123 01\\ &= 23 \cdot 43 \cdot 59 \cdot 49973153089 \cdot 87081167449\\ &\quad \cdot 23574912243518401042300091.\end{aligned}$$

**Beispiel B.1.3** Die Matrix des Hecke-Operators $T_p$ zu der 43 Bit großen Primzahl $p = 8796093039787$ ist

$$B_{T_p} = \begin{pmatrix} 999792 & -430919 & -667507 & -330138 \\ 366631 & 1069844 & 632481 & 993348 \\ -337369 & 29262 & 827492 & -201562 \\ 372395/2 & 1129155/2 & 387026 & 1069844 \end{pmatrix}.$$

Das charakteristische Polynom $\chi_{T_p}$ von $B_{T_p}$ ist

$$\begin{aligned}\chi_{T_p}(x) &= x^4 - 3966972x^3 + 5374646823573x^2\\ &\quad -2838741902427560576x + 4722608337531542138609 77.\end{aligned}$$

Die Ordnung von $J_0(47)_p(\mathbb{F}_p)$ ist

$$\begin{aligned}\chi_{T_p}(p+1) &= 59863080545838468976835845517614339923928552705595 53\\ &= 23 \cdot 229 \cdot 2070421657 \cdot 95568768091\\ &\quad \cdot 57440862681255272137062908 57.\end{aligned}$$

**Beispiel B.1.4** Die Matrix des Hecke-Operators $T_p$ zu der 43 Bit großen Primzahl $p = 8796093040633$ ist

$$B_{T_p} = \begin{pmatrix} -1369823 & -357541 & -1265323 & -219046 \\ 386513 & -108239 & 634531 & -269512 \\ -1046277 & -659764 & -2306577 & -276990 \\ 1677069/2 & 499775/2 & 1347187 & -108239 \end{pmatrix}.$$

Das charakteristische Polynom $\chi_{T_p}$ von $B_{T_p}$ ist

$$\begin{aligned} \chi_{T_p}(x) &= x^4 + 3892878x^3 + 3824316864055x^2 \\ &\quad +1217695474966570526x + 66308346773098377288389. \end{aligned}$$

Die Ordnung von $J_0(47)_p(\mathbb{F}_p)$ ist

$$\begin{aligned} \chi_{T_p}(p+1) &= 59863134060235141443123371568124769247543063235387 01 \\ &= 19 \cdot 23 \cdot 2648562313 \cdot 191956924993 \\ &\quad \cdot 269441218444200793199942580 97. \end{aligned}$$

**Beispiel B.1.5** Die Matrix des Hecke-Operators $T_p$ zu der 43 Bit großen Primzahl $p = 8796093042043$ ist

$$B_{T_p} = \begin{pmatrix} 2908064 & -14683 & 754233 & 333998 \\ 530363 & -24492 & 712045 & 3659284 \\ 420235 & 950598 & 3161300 & -697362 \\ -1886513/2 & 2541687/2 & -1411214 & -24492 \end{pmatrix}.$$

Das charakteristische Polynom $\chi_{T_p}$ von $B_{T_p}$ ist

$$\begin{aligned} \chi_{T_p}(x) &= x^4 - 6020380x^3 + 2591065998317x^2 \\ &\quad +370231710172261482 08x - 53740760428745959873792287. \end{aligned}$$

Die Ordnung von $J_0(47)_p(\mathbb{F}_p)$ ist

$$\begin{aligned} \chi_{T_p}(p+1) &= 59863066632479672058745915372807817157176 10884768753 \\ &= 23 \cdot 30609083173 \cdot 50840027445059191433 \\ &\quad \cdot 167253427730190799379. \end{aligned}$$

## B.2 Beispiele zur Stufe $N = 53$

Die Modulkurve $X_0(53)$ ist eine bielliptische Kurve von Geschlecht $g = 4$. Der Quotient von $X_0(53)$ mit der Atkin-Lehner-Involution $w_{53}$ ist die elliptische Kurve

$$y^2 - xy - y = x^3 - x^2.$$

$X_0(53)$ ist nicht hyperelliptisch und nach N. Elkies wird der Funktionenkörper von $X_0(53)$ dargestellt von $x$, $y$ und einer Wurzel von $f(x, y)$ mit

$$f(x, y) := x^4 - 7x^3 + 9x^2 - 8x - 11 - (2x^2 + 3x - 11)y.$$

Eine Basis für den zu den Spitzenformen gehörigen Raum der Modulsymbole $\mathcal{S}_2(53)$ ist gegeben durch die Modulsymbole

$$m_1 := \{-1/21, 0\} \quad m_2 := \{-1/13, 0\} \quad m_3 := \{-1/35, 0\} \quad m_4 := \{-1/26, 0\}.$$

**Beispiel B.2.1** Die Matrix des Hecke-Operators $T_p$ zu der 43 Bit großen Primzahl $p = 8796093025123$ ist

$$A_{T_p} = \begin{pmatrix} -5524207 & 0 & 0 & 0 \\ 3191271 & -452979 & -639597 & 1311314 \\ -4525359 & -3869702 & -4322681 & -5181016 \\ 4478495 & 1295254 & 1934851/2 & 2137529 \end{pmatrix}.$$

Das charakteristische Polynom $\chi_{T_p}$ von $A_{T_p}$ ist

$$\begin{aligned} \chi_{T_p}(x) &= x^4 + 8162338x^3 + 7162266210134x^2 \\ &\quad -44291241380068260924x - 18503885240506467856404701. \end{aligned}$$

Die Ordnung von $J_0(53)_p(\mathbb{F}_p)$ ist

$$\begin{aligned} \chi_{T_p}(p+1) &= 59863162694454335876243931149411019855175878979093 95 \\ &= 3^5 \cdot 5 \cdot 13^2 \cdot 839 \cdot 5227 \cdot 565567 \cdot 36197936417 \\ &\quad \cdot 32472396435632831952711 1. \end{aligned}$$

**Beispiel B.2.2** Die Matrix des Hecke-Operators $T_{p'}$ zu der 43 Bit großen Primzahl $p' = 8796093030211$ ist

$$A_{T_{p'}} = \begin{pmatrix} 829037 & 0 & 0 & 0 \\ -1174937 & -3973191 & 425005 & 2452354 \\ -1978511 & -752334 & -4725525 & -3204688 \\ -1199356 & 801172 & 376167/2 & -2370847 \end{pmatrix}.$$

Das charakteristische Polynom $\chi_{T_{p'}}$ von $A_{T_{p'}}$ ist

$$\begin{aligned}\chi_{T_{p'}}(x) &= x^4 + 10240526x^3 + 29179398120126x^2 \\ &\quad +8021305065282605760x - 330124542810317941478652 53.\end{aligned}$$

Die Ordnung von $J_0(53)_{p'}(\mathbb{F}_{p'})$ ist

$$\begin{aligned}\chi_{T_{p'}}(p'+1) &= 59863176976394671641013978400705875590823384418172 75 \\ &= 3 \cdot 5^2 \cdot 13 \cdot 41 \cdot 5849 \cdot 489061 \\ &\quad \cdot 523511993973852178208998715086433672 81\end{aligned}$$

Der letzte Faktor ist eine 125-Bit Zahl.

# Literaturverzeichnis


[Ala87] V. S. Alagar, D. K. Probst, *A fast, low-space algorithm for multiplying dense multivariate polynomials*, ACM Transactions on Mathematical Software 3(1):35-57, 1987

[AtB99] A. Atkin, D.J. Bernstein, *Prime sieves using binary quadratic forms*, http://cr.yp.to/papers/primesieves-19990826.pdf, 1999

[AtL70] A. Atkin, L. Lehner, *Hecke operators on* $\Gamma_0(m)$., Math. Ann. 185:134-160, 1970

[AtM93] A. Atkin, F. Morain, *Elliptic curves and primality proving*, Math. Comp. 61(203):29-68, 1993

[BaP98] D. Baley, C. Paar, *Optimal extension fields for fast arithmetic in public-key algorithms.*, Advances in Cryptology - CRYPTO '98, Lecture Notes in Comput. Sci. 1462:472-485, Springer-Verlag, 1998

[Bas96] J. Basmaji, *Ein Algorithmus zur Berechnung von Hecke-Operatoren und Anwendung auf modulare Kurven*, Dissertation Universität Duisburg-Essen, 1996

[BCS05] A. Bostan, T. Cluzeau, B. Salvy, *Fast algorithms for Polynomial Solutions of Linear Differential Equations*, In M. Kauers (ed), Symbolic and Algebraic Computation. - New York, 2005. Proceedings of ISSAC'05

[BGS04] A. Bostan, P. Gaudry, E. Schost, *Linear recurrences with polynomial coefficients and computation of the Cartier-Manin operator on hyperelliptic curves*, 2004
http://www.lix.polytechnique.fr/Labo/Pierrick.Gaudry/publis/factor.ps.gz

[BGS07] A. Bostan, P. Gaudry, E. Schost, *Linear Recurrences with Polynomial Coefficients and Application to Integer Factorization and Cartier-Manin Operator*, SIAM Journal on Computing, 36(6):1777-1806

[Bou98] I. Bouw, *Tame covers of curves: p-ranks and fundamental groups*, Dissertation, Universität Utrecht, 1998

[Bri88] D. Le Brigand, J.J. Risler, *Algorithme de Brill-Noether et codes de Goppa* Bull. Soc. math. France 116:231-253, 1988

[BTW05] M. Bauer, E. Teske, A. Weng, *Point counting on Picard curve in large characteristic*, Mathematics of Computation 74:1983-2005, 2005, http://www.mathematik.uni-mainz.de/∼weng/pc.ps

[CDV06] W. Castryck, J. Denef, F. Vercauteren, *Computing Zeta Functions of Nondegenerate Curves*, International Mathematics Research Papers, vol. 2006, Article ID 72017, 57 pages, 2006

[Chu88] D. V. Chudnovsky, G. V. Chudnovsky, *Approximations and complex multiplication according to Ramanujan.*Academic Press, Boston, MA, Ramanujan Revisited (Urbana-Champaign, Ill., 1987) S. 375-472, 1988

[CMT02] J. Chao, K. Matsuo, S. Tsujii, *An improved baby step giant step algorithm for point counting of hyperelliptic curves over finite fields*, ANTS-V (C. Fieker and D.R. Kohel, eds.), Springer-Verlag, Lecture Notes in Computer Science 2369:461-474, 2002

[CoF+06] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography.* Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2006.

[DFK+97] M. Daberkow, C. Fieker, J. Klüners, M. Pohst, K. Roegner, M. Schörnig, K. Wildanger, *KANT V4*, J. Symb. Comp., 24(3):267-283, 1997

[DeV06] J. Denef, F. Vercauteren, *Computing Zeta functions of Cab curves using Monsky-Washnitzer cohomology*, Finite Fields Appl. 12(1):78-102, 2006, errata: http://www.wis.kuleuven.be/algebra/denef_papers/ErrataPointCounting.pdf

[Die05] C. Diem, *Index Calculus in Class Groups of Plane Curves of Small Degree*, http://citeseer.ist.psu.edu/724849.html, 2005

[Elk98] N. Elkies, *Elliptic and modular curves over finite fields and related computational issues*, Computational Perspectives on Number Theory: Proceedings of a Conference in Honor of A.O.L. Atkin, AMS/International Press, S. 21-76, 1998

[Est91] J. Estrada-Sarlabous, *On the Jacobian Varieties of Picard Curves Defined over Fields of Characteristic* $p > 0$, Math. Nachr., 152:329-340, 1991

[FlS97] P. Flajolet, B. Salvy, *The SIGSAM challenges: Symbolic asymptotics in practice.*, SIGSAM Bull., 31(4):36-47, 1997

[FrMü98] G. Frey, M. Müller, *Arithmetic of Modular Curves and Applications*, Algorithmic algebra and number theory, Ed. Matzat et al., Springer-Verlag, Berlin, S. 11-48. MR 00a:11095, 1998

[GaG01] P. Gaudry, N. Gürel, *An extension of Kedlaya's point-counting algorithm to superelliptic curves*, Advances in cryptology-ASIACRYPT 2001 (Gold Coast), Lecture Notes in Comput. Sci., 2248:480-494, Springer-Verlag, Berlin, 2001

[GaG03] P. Gaudry, N. Gürel, *Counting points in medium characteristic using Kedlaya's algorithm*, Experimental Math. 12:395-402, 2003

[Gal96] S. Galbraith, *Equations for Modular Curves* Dissertation, Oxford, 1996

[GaH00] P. Gaudry, R. Harley, *Counting points on hyperelliptic curves over finite fields*, ANTS-IV, Springer-Verlag, LNCS 1838, 313-332, 2000.

[GaS04] 3. P. Gaudry, É. Schost, *A Low-Memory Parallel Version of Matsuo, Chao and Tsujii's Algorithm.*, ANTS VI, LNCS 3076, S. 208-222, Springer-Verlag, 2004

[Gor52] D. Gorenstein, *An arithmetic theory of adjoint curves*, Trans. Amer. Math. Soc. 72:414-436, 1952

[Hac96] G. Haché, *Construction effective des codes géométriques*, Ph.D. Thesis, Univ. Paris 6, 1996

[Har77] R. Hartshorne, *Algebraic Geometry*, Springer Verlag, 1977

[Hav06] D. Harvey, *Kedlaya's Algorithm in Larger Characteristic*, eprint arXiv:math/0610973, 2006

[HaW36] H. Hasse, E. Witt, *Zyklische unverzweigte Erweiterungskörper vom Primzahlgrade p über einem algebraischen Funktionenkörper der Charakteristik p*, Monatsh. Math. Phys., 43:477-492, 1936

[Hes02] F. Hess, *Computing Riemann-Roch Spaces in algebraic function fields and related topics*, Journal of Symbolic Computation 33(4):425-445, 2002

[Ler03] R. Lercier, D. Lubicz, *Algorithmic aspects of Mestre's p-adic point counting ideas*, ECC 2003 Slides, http://www.cacr.math.uwaterloo.ca/conferences/2003/ecc2003/lercier.pdf

[Magma] *Magma HTML Help Document*, http://www.msri.org/about/computing/docs/magma/html/magma.htm

[Man65] Ju. I. Manin,*The Hasse-Witt Matrix of an Algebraic Curve* Transl. Amer. Math. Soc. 45:245-264, 1965

[Man72] J. Manin, *Parabolic Points und Zeta-Functions of Modular Curves*, Math. USSR Izvestija, 6(1):19-64, 1972

[Maz77] B. Mazur, *Modular Curves and the Eisenstein ideal.*, Publ. Math. IHES, 47:33-186, 1977

[Mer94] L. Merel, *Universal Fourier expansions of modular Forms*, In G. Frey editor, On Artin's Conjecture for Odd 2-dimensional Representations, Lecture Notes in Mathematics 1585:59-94, Berlin, Heidelberg, Springer-Verlag, 1994

[Mum74] D. Mumford, *Abelian Varieties.*, Oxford University Press, Oxford, 2nd edition, 1974

[Neu92] J. Neukirch, *Algebraische Zahlentheorie*, Springer Verlag, 1992

[Ogg78] A. Ogg, *On the Weierstrass Points of $X_0(N)$*, Illinois J. Math., 22(1):31-35, 1978

[Pil90] J. Pila, *Frobenius maps of abelian varieties and finding roots of unity in finite fields*. Math. Comp., 1990

[PPW03] C. Paar, J. Pelzl, T. Wollinger, *Low Cost Security: Explicit Formulae for Genus 4 Hyperelliptic Curves*, Tenth Annual Workshop on Selected Areas in Cryptography - SAC, http://citeseer.ist.psu.edu/584454.html, 2003

[Rit04] C. Ritzenthaler, *Point counting on genus 3 non hyperelliptic curves*, Algorithmic Number Theory 6th International Symposium, ANTS VI, University of Vermont 13-18 Proceedings, 2004

[Roh97] D. Rohrlich, *Modular Curves, Hecke Correspondences and L-Functions*. In G. Cornell, J.H. Silverman and G. Stevens Editors, Modular Forms and Fermat's Last Theorem, S. 41-100, Berlin, Heidelberg, Springer-Verlag, 1997

[Shi73] G. Shimura, *On modular forms of half integral weight*, Annals of Mathematics, 97:440-481, 1973

[Shi94] G. Shimura, *Introduction to the arithmetic theory of automorphic functions*, Princeton University Press, Princeton, NJ, 1994

[Sch74] B. Schoeneberg, *Elliptic modular functions*, Springer-Verlag New York, Grundlehren der math. Wissenschaften 203, 1974

[ScS03] R. Scheidler, A. Stein, *Computing the class number of cubic function fields*, Vortrag von A. Stein bei „Conference in Number Theory in Honor of Professor H.C. Williams", Banff, 2003

[Ser58] J. P. Serre, *Sur la topologie des variété algébriques en characteristique p*, Symposium International de Topología Algebraica UNAM, UNESCO 24-53, 1958

[SGA4] *Théorie des topos et cohomologie étale des schémas.*, Springer-Verlag, Berlin, 1973, Séminaire de Géométrie Algébrique du Bois-Marie 1963-1964(SGA 4), Dirigé par M. Artin, A. Grothendieck et J. L. Verdier. Lecure Notes in Math., Vol. 269, 270, 305.

[Sil86] J. H. Silverman, *The Arithmetic of Elliptic Curves*, Springer Verlag, 1986

[Sti93] H. Stichtenoth, *Algebraic Function Fields and Codes*, Springer Verlag, 1993

[StV87] K.-O. Stöhr, J. F. Voloch, *A formula for the Cartier operator on plane algebraic curves*, J. Reine Angew. Math. (Crelle's Journal), 377:49-64, 1987

[Tat52] J. Tate, *Genus change in inseparable extensions of function fields*, Proc. Amer. Math. Soc. 3:400-406, 1952

[Tei07] X. Teixés, Dissertation, Universität Duisburg-Essen, erscheint 2007

[Ten95] G. Tenenbaum, *Introduction to Analytic and Probabilistic Number Theory*, Cambridge University Press, Cambridge Studies in Advanced Mathematics 46, 1995

[Web97] H.-J. Weber, *Algorithmische Konstruktion hyperelliptischer Kurven mit kryptographischer Relevanz und einem Endomorphismenring echt größer als* $\mathbb{Z}$, Dissertation Universität Duisburg-Essen, 1997

[Yui78] N. Yui, *On the jacobian varieties of hyperelliptic curves over fields of characteristic* $p > 0$., J. Algebra 52:378-410, 1978